




CANopen S300/S700

Kommunikationsprofil

Feldbus Schnittstelle
Originalanleitung
Ausgabe 12/2009

Bewahren Sie das Handbuch als Produktbestandteil während der Lebensdauer des Produktes auf.
Geben Sie das Handbuch an nachfolgende Benutzer oder Besitzer des Produktes weiter.

Datei s300700can_d.***



Bisher erschienene Ausgaben

Ausgabe	Bemerkung
10 / 2005	Erstausgabe (gültig ab Firmware 1.32 - CAN Version 0.41)
11 / 2005	Kombiniert: Referenz und Beispiele
09 / 2006	Neues Design, SDO umbenannt in Objekt, Index erweitert
11 / 2006	Aktualisierung für S700 Familie, Abschlusswiderstände (Interface) korrigiert, neue Objekte
08 / 2007	Symbole, Normen
01 / 2008	Korrekturen, neue Objekte,Symbole nach ANSI Z535
12 / 2008	Kleine Korrekturen
07 / 2009	Statusmaschine Bit 6 korrigiert, CAN Logo, product brand
12 / 2009	Kleinere Korrekturen

Technische Änderungen, die der Verbesserung der Geräte dienen, vorbehalten!

Gedruckt in der BRD

Seite

Seite

Seite

Seite

1 Allgemeines

1.1 Über dieses Handbuch

Dieses Handbuch beschreibt Inbetriebnahme, Funktionsumfang und Softwareprotokoll des Kommunikationsprofils CANopen bei Servoverstärkern der Serien S300/S700. Es ist Bestandteil der Gesamtdokumentation der digitalen Servoverstärker S300/S700.

Installation und Inbetriebnahme der Servoverstärker, sowie alle Standardfunktionen werden in der zugehörigen Betriebsanleitung beschrieben.

Sonstige Bestandteile der Gesamtdokumentation der digitalen Servoverstärker-Familien:

Titel	Herausgeber
Betriebsanleitung S300/S700	Kollmorgen
Online-Hilfe mit Objekt Referenz in der Inbetriebnahme-Software	Kollmorgen

Weiterführende Dokumentation:

Titel	Herausgeber
CAN Application Layer (CAL) for Industrial Applications	CiA e.V.
Draft standards 301 (ab Version 4.0), 402	CiA e.V.
CAN Specification Version 2.0	CiA e.V.
ISO 11898 ...Controller area network (CAN) for high-speed communication	

1.2 Zielgruppe

Dieses Handbuch richtet sich mit folgenden Anforderungen an Fachpersonal:

Transport:	nur durch Personal mit Kenntnissen in der Behandlung elektrostatisch gefährdeter Bauelemente
Auspacken:	nur durch Fachleute mit elektrotechnischer Ausbildung
Installation:	nur durch Fachleute mit elektrotechnischer Ausbildung
Inbetriebnahme:	nur durch Fachleute mit weitreichenden Kenntnissen in den Bereichen Elektrotechnik / Antriebstechnik
Programmierung:	Softwareentwickler, CAN-BUS Projekteure

Das Fachpersonal muss folgende Normen kennen und beachten:

IEC 60364 und IEC 60664

nationale Unfallverhütungsvorschriften

⚠ WARNUNG **Während des Betriebes der Geräte besteht die Gefahr von Tod oder schweren gesundheitlichen oder materiellen Schäden. Der Betreiber muss daher sicherstellen, dass die Sicherheitshinweise in diesem Handbuch beachtet werden. Der Betreiber muss sicherstellen, dass alle mit Arbeiten am Servoverstärker betrauten Personen die Betriebsanleitung gelesen und verstanden haben.**

Wir bieten auf Anfrage Schulungs- und Einarbeitungskurse an.

1.3 Hinweise für die Online-Ausgabe (PDF-Format)

Lesezeichen:
Inhaltsverzeichnis und Index sind aktive Lesezeichen.
Inhaltsverzeichnis und Index im Text:
Die Zeilen sind aktive Querverweise. Klicken Sie auf die gewünschte Zeile und die entsprechende Seite wird angezeigt.
Seitenzahlen im Text:
Seitenzahlen/Kapitelzahlen bei Querverweisen sind aktiv. Klicken Sie auf die Seitenzahl/Kapitelzahl um zum angegebenen Ziel zu gelangen.

1.4 Bestimmungsgemäße Verwendung des CANopen Interfaces

Beachten Sie das Kapitel "Bestimmungsgemäße Verwendung" in der Betriebsanleitung des Servoverstärkers.

Das CANopen Interface dient allein dem Anschluss des Servoverstärkers an einen Master mit CAN BUS Anbindung.
Die Servoverstärker werden als Komponenten in elektrischen Anlagen oder Maschinen eingebaut und dürfen nur als integrierte Komponenten der Anlage in Betrieb genommen werden.

INFO

Wir garantieren nur bei Verwendung der von uns genannten Komponenten und Einhaltung der Installationsvorschriften die Konformität der Servoverstärker zu folgenden Normen im Industriebereich:

EG-EMV-Richtlinie	**2004/108/EG**
EG-Niederspannungs-Richtlinie	**2006/95/EG**

1.5 Symbole

Symbol	Bedeutung
⚠ GEFAHR	Weist auf eine gefährliche Situation hin, die, wenn sie nicht vermieden wird, zum Tode oder zu schweren, irreversiblen Verletzungen führen wird.
⚠ WARNUNG	Weist auf eine gefährliche Situation hin, die, wenn sie nicht vermieden wird, zum Tode oder zu schweren, irreversiblen Verletzungen führen kann.
⚠ VORSICHT	Weist auf eine gefährliche Situation hin, die, wenn sie nicht vermieden wird, zu leichten Verletzungen führen kann.
HINWEIS	Weist auf eine Situation hin, die, wenn sie nicht vermieden wird, zu Beschädigung von Sachen führen kann.
INFO	Dies ist kein Sicherheits-Symbol. Dieses Symbol weist auf wichtige Informationen hin.

1.6 Kürzel

In der Tabelle unten werden die in diesem Handbuch verwendeten Abkürzungen erklärt.

Kürzel	Bedeutung
BTB/RTO	Betriebsbereit
COB	Communication Object
COB-ID	Communication Object Identifier
EEPROM	Elektrisch löschbarer Festspeicher
EMV	Elektromagnetische Verträglichkeit
ISO	International Standardization Organisation
LED	Leuchtdiode
MB	Megabyte
NSTOP	Endschaltereingang Drehrichtung links
PC	Personal Computer
PDO	Prozessdatenobjekt
PSTOP	Endschaltereingang Drehrichtung rechts
RAM	flüchtiger Speicher
ROD	Inkrementelle Positionsausgabe
RXPDO	Receive- (Empfangs-) PDO
SDO	Servicadatenobjekt
TXPDO	Transmit- (Sende-) PDO

1.7 Realisierte Grundfunktionen über CANopen

In Zusammenhang mit dem im digitalen Servoverstärker S300/S700 integrierten Lageregler werden folgende Funktionen bereitgestellt:

Einricht- und Allgemeine Funktionen:

— Referenzfahren, Referenzpunkt setzen
— Vorgabe digitaler Sollwerte für die Drehzahl- und Momentenregelung
— Unterstützung folgender Betriebsarten des CANopen Profils DS402:
» profile position mode
» homing mode
» profile torque mode
» interpolated position mode
» profile velocity mode

Positionierfunktionen:

— Ausführen eines Fahrauftrages aus dem Fahrsatzspeicher des Servoverstärkers
— Ausführen eines Direktfahrauftrages
— Trajektorie absolut, ip-Mode

Datentransferfunktionen:

— Übertragen eines Fahrauftrages in den Fahrsatzspeicher des Servoverstärkers
Ein Fahrauftrag besteht aus folgenden Elementen:
» Positionssollwert (Absolutauftrag) oder Wegsollwert (Relativauftrag)
» Geschwindigkeitssollwert
» Beschleunigungszeit, Bremszeit
» Fahrauftragsart (absolut/relativ)
» Nummer eines Folgefahrauftrags (mit oder ohne Zwischenstopp)
— Lesen eines Fahrauftrages aus dem Fahrsatzspeicher des Servoverstärkers
— Lesen von Istwerten
— Lesen der Fehlerregister
— Lesen der Statusregister
— Lesen / Schreiben von Regelparametern

1.8 Übertragungsgeschwindigkeit und -verfahren

— Busankopplung und Busmedium: CAN-Standard ISO 11898 (CAN-Highspeed)
— Übertragungsgeschwindigkeit: max. 1MBit/s
Einstellmöglichkeiten des Servoverstärkers:
10, 20, 50, 100, 125, 250, 333, 500(default), 666, 800, 1000kBaud

1.9 Verhalten bei Kommunikationsstörungen BUSOFF

Die Kommunikationsstörung BUSOFF wird direkt von der Schicht 2 (CAN controller) überwacht und gemeldet. Diese Meldung kann unterschiedliche Ursachen haben.

Hier einige Beispiele:
- Telegramm wird gesendet, obwohl kein weiterer CAN – Knoten angeschlossen ist
- CAN Knoten weisen unterschiedliche Baudraten auf
- Busleitung defekt
- Reflexionen auf den Leitungen aufgrund fehlerhafter Leitungsabschlüsse.

Ein BUSOFF wird vom S300/S700 nur gemeldet, wenn ein weiterer CAN Knoten angeschlossen ist und mindestens ein Objekt zu Beginn erfolgreich abgesetzt werden konnte. Der Zustand BUSOFF wird mit der Fehlermeldung F23 signalisiert. Sollte beim Auftreten dieses Fehlers die Endstufe freigegeben (enable) sein und eine Fahrfunktion ausgeführt werden, wird der Antrieb mit der Notbremsrampe angehalten und die Endstufe gesperrt (disable).

2 Installation / Inbetriebnahme

2.1 Montage, Installation

GEFAHR

Installieren und verdrahten Sie die Geräte immer in spannungsfreiem Zustand. Weder die Leistungsversorgung, noch die 24V-Hilfsspannung, noch die Betriebsspannung eines anderen anzuschliessenden Gerätes darf eingeschaltet sein.
Sorgen Sie für eine sichere Freischaltung des Schaltschrankes (Sperre, Warnschilder etc.). Erst bei der Inbetriebnahme werden die einzelnen Spannungen eingeschaltet.

In ungünstigen Fällen können Lichtbögen entstehen und Personen und elektrische Kontakte schädigen. Lösen Sie daher die elektrischen Anschlüsse der Servoverstärker nie unter Spannung. Restladungen in den Kondensatoren können auch mehrere Minuten nach Abschalten der Netzspannung gefährliche Werte aufweisen. Messen Sie die Spannung im Zwischenkreis und warten Sie, bis die Spannung unter 40V abgesunken ist. Steuer- und Leistungsanschlüsse können Spannung führen, auch wenn sich der Motor nicht dreht.

VORSICHT

Montieren Sie den Servoverstärker wie in den Betriebsanleitungen S300/S700 beschrieben. Beachten Sie alle Sicherheitshinweise in der zum Servoverstärker gehörenden Betriebsanleitung. Beachten Sie alle Hinweise zu Einbaulage, Umgebungsbedingungen und Verdrahtung, sowie Absicherung.

2.2 Einstellen der Stationsadresse

Bei der Inbetriebnahme ist es sinnvoll, die Stationsadressen der einzelnen Verstärker und die Baudrate für die Kommunikation vorab über die Frontplattentastatur einzustellen (siehe Kapitel "Inbetriebnahme" in der Betriebsanleitung)

INFO

Nach Verändern der Stationsadresse und Baudrate müssen Sie die 24V-Hilfsspannungs-Versorgung der Servoverstärker aus- und wieder einschalten.

Die **Stationsadresse** (Bereich 1...127) können Sie auf drei Arten einstellen:

- Mit der Tastatur in der Frontplatte (siehe Kapitel "Inbetriebnahme" in der Betriebsanleitung)
- In der Inbetriebnahme-Software DriveGUI.exe auf der Bildschirmseite "CAN / Feldbus”
- Über die serielle Schnittstelle mit der Abfolge der ASCII-Kommandos:
  ADDR nn ⇒ SAVE ⇒ COLDSTART (mit nn = Adresse)

Die CAN - **Übertragungsgeschwindigkeit** (Baudrate) können Sie auf drei Arten einstellen:

- Mit der Tastatur in der Frontplatte (siehe Kapitel "Inbetriebnahme" in der Betriebsanleitung)
- In der Inbetriebnahme-Software DriveGUI.exe auf der Bildschirmseite "CAN / Feldbus”
- Über die serielle Schnittstelle mit der Abfolge der ASCII - Kommandos:
  CBAUD bb ⇒ SAVE ⇒ COLDSTART (mit bb = Baudrate in kB)

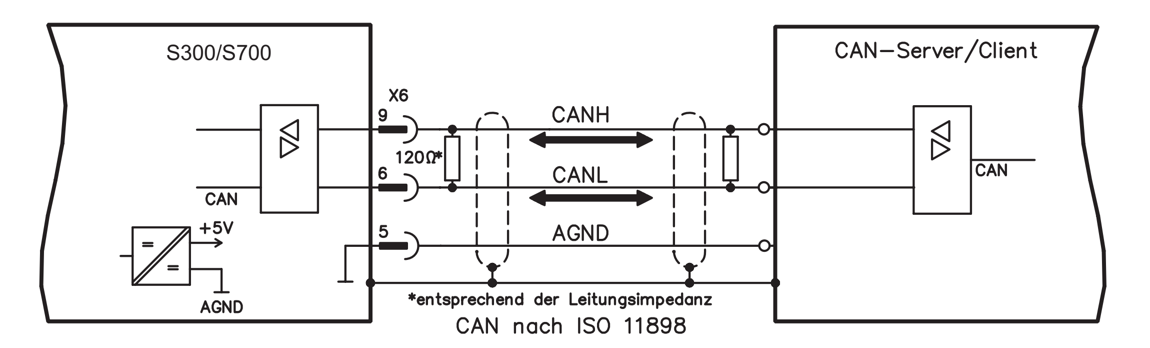
Codierung der Baudrate im LED-Display :

Kodierung	Baud Rate in kbit/s	Kodierung	Baud Rate in kbit/s
1	10	33	333
2	20	50	500
5	50	66	666
10	100	80	800
12	125	100	1000
25	250		

2.3 CANopen Interface (X6)

Interface zum Anschluss an den CAN Bus (default : 500 kBaud). Die Schnittstelle liegt auf dem gleichen Potential wie das RS232-Interface. Die analogen Sollwerteingänge sind weiterhin nutzbar.

Wir liefern auf Anfrage spezielle Klemmbuchsenstecker, die für den Busbetrieb leicht konfektioniert werden können.



2.4 CAN Busleitung

Nach ISO 11898 sollten Sie eine Busleitung mit einem Wellenwiderstand von 120 Ω verwenden. Die verwendbare Leitungslänge für eine sichere Kommunikation nimmt mit zunehmender Übertragungsrate ab. Als Anhaltspunkte können folgende bei uns gemessenen Werte dienen, die allerdings nicht als Grenzwerte zu verstehen sind:

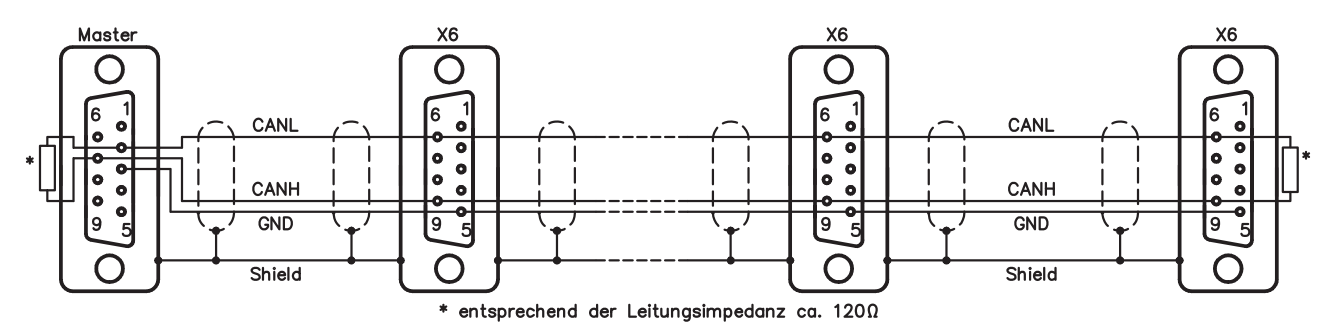
Leitungsdaten:	Wellenwiderstand	100..120 Ω
	Leitungskapazität	max. 60 nF/km
	Leiterwiderstand (Schleife)	159,8 Ω/km

Leitungslängen in Abhängigkeit von der Übertragungsrate

Übertragungsrate / kBaud	max. Leitungslänge / m
1000	10
500	70
250	115

Mit geringerer Leitungskapazität (max. 30 nF/km) und geringerem Leiterwiderstand (Schleife, 115 Ω/km) können größere Übertragungsweiten erreicht werden.
(Wellenwiderstand 150 ± 5Ω ⇒ Abschlusswiderstand 150 ± 5Ω).

An das SubD-Steckergehäuse sind aus EMV-Gründen folgende Anforderung gestellt:
— metallisches oder metallisch beschichtetes Gehäuse
— Anschlussmöglichkeit für den Leitungsschirm im Gehäuse, großflächige Verbindung



2.5 Leitfaden zur Inbetriebnahme

⚠VORSICHT **Nur Fachpersonal mit fundierten Kenntnissen in Regelungstechnik und Antriebstechnik darf den Servoverstärker in Betrieb nehmen.**

Montage / Installation prüfen	Prüfen Sie, ob alle Sicherheitshinweise in der Betriebsanleitung des Servoverstärkers und im vorliegenden Handbuch beachtet und umgesetzt wurden. Kontrollieren Sie Adresse und Baudrate.
PC anschließen, Inbetriebnahmesoftware starten	Zum Parametrieren des Servoverstärkers verwenden Sie die Inbetriebnahmesoftware DriveGUI.exe.
Grundfunktionen in Betrieb nehmen	Nehmen Sie nun die Grundfunktionen des Servoverstärkers in Betrieb und optimieren Sie Strom- und Drehzahlregler. Dieser Teil der Inbetriebnahme ist in der Betriebsanleitung und in der Onlinehilfe der Inbetriebnahmesoftware genauer beschrieben.
Parameter speichern	Speichern Sie die Parameter nach erfolgter Optimierung im Servoverstärker.
Buskommunikation in Betrieb nehmen	Die geänderten Parameter werden erst nach einem Software-Reset (Warmboot) wirksam. Betätigen Sie dazu den Reset-Button in der Symbolleiste der Inbetriebnahmesoftware. Es ist erforderlich, dass das in Kapitel 4 beschriebene Software-Protokoll ist auf dem Master realisiert ist. Baudrate des S300/S700 an die des Masters anpassen.
Test der Kommunikation	Prüfen Sie die Boot-Up Meldung, wenn Sie den Verstärker einschalten. Führen Sie einen SDO Lesezugriff auf Index 1000 Subindex 0 aus (Gerätetyp). Siehe auch Beispiele im Anhang ab S.102

⚠WARNUNG **Stellen Sie sicher, dass auch bei ungewollter Bewegung des Antriebs keine maschinelle oder personelle Gefährdung eintreten kann.**

Lageregler in Betrieb nehmen	Nehmen Sie den Lageregler in Betrieb, wie in der Online-Hilfe der Inbetriebnahmesoftware beschrieben.

2.6 Wichtige Konfigurationsparameter für den CAN-Bus - Betrieb

Die folgenden Parameter sind für den CAN - Betrieb wichtig:

1. CBAUD : Übertragungsrate für den CAN-Bus

2. ADDR : Mit dem Kommando ADDR wird die Feldbus-Adresse des Verstärkers definiert. Nach der Änderung der Adresse sollten alle Parameter im EEPROM abgespeichert werden und der Verstärker aus- und eingeschaltet werden.

3. AENA: Hiermit kann der Zustand des Software-Enable beim Einschalten des Verstärkers definiert werden. Mit dem Software-Enable wird einer externen Steuerung die Möglichkeit gegeben die Endstufe softwaremässig zu sperren bzw. freizugeben. Bei Geräten, die mit einem analogen Sollwert arbeiten (OPMODE=1,3) wird beim Einschalten des Verstärkers das Software-Enable automatisch gesetzt, so dass diese Geräte sofort betriebsbereit sind (Hardware-Enable vorausgesetzt). Bei allen anderen Geräten wird beim Einschalten das Software-Enable auf den Wert von AENA gesetzt. Die Variable AENA hat auch eine Funktion beim Resettieren des Verstärkers nach einem Fehler (über digitalen Eingang 1 bzw. mit dem ASCII-Kommando CLRFAULT). Bei Fehlern, die softwaremässig resettiert werden können, wird, nachdem der Fehler gelöscht wurde, der Software-Enable auf den Zustand von AENA gesetzt. Auf diese Weise ist das Verhalten des Verstärkers beim Software-Reset analog zu dem Einschaltverhalten.

3 CANopen Kommunikationsprofil

Dieses Kapitel beschreibt die grundlegenden Dienste und Kommunikationsobjekte des CANopen Kommunikationsprofils DS 301, die vom S300/S700 verwendet werden.

INFO

Es wird vorausgesetzt, dass das Kommunikationsprofil in seiner grundlegenden Funktionsweise bekannt ist und als Referenzdokumentation zur Verfügung steht.

Da die Geschäftssprache der CiA englisch ist, werden feststehende Begriffe im Original verwendet (z.B. Objekt 1000_h: Device Type).

3.1 Allgemeine Erläuterungen zu CAN

Das hier verwendete Übertragungsverfahren ist in der ISO 11898 (Controller Area Network [CAN] for high-speed communication) definiert.
Das in allen CAN-Bausteinen implementierte Schicht 1/2-Protokoll (Physical Layer/Data Link Layer) stellt u. a. die Anforderung von Daten zur Verfügung.

Datentransport bzw. Datenanforderung erfolgen über ein Datentelegramm (Data Frame) mit bis zu 8 Byte Nutzdaten bzw. über ein Datenanforderungstelegramm (Remote Frame).

Kommunikationsobjekte (COBs) werden durch einen 11 Bit Identifier (ID) gekennzeichnet, der auch die Priorität von Objekten bestimmt.

Um die Applikation von der Kommunikation zu entkoppeln, wurde ein Schicht-7-Protokoll (Anwendungsschicht) entwickelt. Die von der Anwendungsschicht bereitgestellten Dienstelemente ermöglichen die Realisierung einer über das Netzwerk verteilten Applikation. Diese Dienstelemente sind im "CAN Applikation Layer (CAL) for Industrial Applications" beschrieben.
Auf die CAL ist das Kommunikationsprofil CANopen und das Antriebsprofil aufgesetzt.

Das folgende Diagramm zeigt die grundlegende Struktur eines Kommunikationsobjekts:

S O M	COB-ID	R T R	CTRL	Data Segment	CRC	A C K	EOM

SOM	Start of message	
COB-ID	Kommunikationsobjekt-Identifier (11 Bit)	
RTR	Remote Transmission Request	
CTRL	Control Field (u.a. Data Length Code)	
Data Segment	0..8	Byte (Data-COB)
	0	Byte (Remote-COB)
CRC	Prüfsequenz	
ACK	Acknowledge Slot	
EOM	End of message	

3.2 Aufbau des Kommunikationsobjekt-Identifiers

Die folgende Grafik zeigt den Aufbau des COB – Identifiers (COB - ID). Der "Function Code" legt die Bedeutung und die Priorität des jeweiligen Objekts fest.

10	9	8	7	6	5	4	3	2	1	0
Function-Code				Module-ID						

Bit 0..6 Modul ID (CAN-Bus Adresse des Servoverstärkers, Bereich 1..127; wird in der Inbetriebnahmesoftware oder am Servoverstärker eingestellt, ⇒ S.10)

Bit 7..10 Function Code (Nummer des im Server definierten Kommunikationsobjektes)

INFO **Wird eine ungültige Stationsnummer eingestellt (=0 oder >127), so wird die Modul-ID intern auf 1 gesetzt.**

Die folgenden Tabellen zeigen die Default - Belegung der COB - Identifier nach dem Einschalten des Servoverstärkers. Die Objekte, die mit einem Index (Communication Parameters at Index) versehen sind, können nach der Initialisierungsphase mit einem neuen Identifier versehen werden. Die Indizes in Klammern sind optional.

Vordefinierte "Broadcast" Objekte (Senden an alle)

Objekt	Function code (binary)	Resulting COB-IDs		Communication parameters at index
		Dez.	Hex.	
NMT	0000	0	0_h	—
SYNC	0001	128	80_h	(1005_h)
TIME	0010	256	100_h	nicht unterstützt

Vordefinierte "Peer to Peer" Objekte (Senden Station zu Station)

Objekt	Function code (binary)	Resulting COB-IDs		Communication parameters at index	Piorität
		Dez.	Hex.		
EMERGENCY	0001	129..255	81_h..FF_h	—	hoch
TPDO 1	0011	385..511	181_h..$1FF_h$	1800_h	
RPDO 1	0100	513..639	201_h..$27F_h$	1400_h	
TPDO 2	0101	641..767	281_h..$2FF_h$	1801_h	
RPDO 2	0110	769..895	301_h..$37F_h$	1401_h	
TPDO 3	0110	897..1023	381_h..$3FF_h$	1802_h	
RPDO 3	1000	1025..1151	401_h..$47F_h$	1402_h	
TPDO 4	1001	1153..1279	481_h..$4FF_h$	1803_h	
RPDO 4	1010	1281..1407	501_h..$57F_h$	1403_h	
SDO (tx*)	1011	1409..1535	581_h..$5FF_h$		
SDO (rx*)	1100	1537..1663	601_h..$67F_h$		↓
Nodeguard	1110	1793..1919	701_h..$77F_h$	($100E_h$)	niedrig

* tx = Senderichtung S300/S700 ⇒ Master
rx = Senderichtung Master ⇒ S300/S700

3.3 Definition der verwendeten Datentypen

Dieses Kapitel definiert die verwendeten Datentypen. Jeder Datentyp kann mit Hilfe von Bit -Sequenzen beschrieben werden. Diese Bit - Sequenzen werden in "Octets" (Bytes) zusammengefasst. Für numerische Datentypen wird das **"Little – Endian" – Format** (Intel) verwendet (s. auch DS301 Application Layer "General Description of Datentyps and Encoding Rules").

3.3.1 Basisdatentypen

3.3.1.1 Unsigned Integer

Daten vom Basis Datentyp UNSIGNEDn definiert ausschließlich positive Integer.
Der Wertebereich ist $0..2^{n}-1$. Die Bitsequenz $b = b_0..b_{n-1}$ definiert den Wert

$$UNSIGNEDn(b) = b_{n-1}\, 2^{n-1} +..+ b_1\, 2^1 + b_0\, 2^0$$

Beispiel: Der Wert 266 = $10A_h$ wird mit dem Datentyp UNSIGNED16 mit Hilfe von zwei "octets" übertragen (1. octet = $0A_h$, 2. octet = 01_h).

Übertragungssyntax für den Datentyp UNSIGNEDn:

octet Nummer	1.	2.	3.	4.	5.	6.	7.	8.
UNSIGNED8	$b_7..b_0$							
UNSIGNED16	$b_7..b_0$	$b_{15}..b_8$						
UNSIGNED24	$b_7..b_0$	$b_{15}..b_8$	$b_{23}..b_{16}$					
UNSIGNED32	$b_7..b_0$	$b_{15}..b_8$	$b_{23}..b_{16}$	$b_{31}..b_{24}$				
UNSIGNED40	$b_7..b_0$	$b_{15}..b_8$	$b_{23}..b_{16}$	$b_{31}..b_{24}$	$b_{39}..b_{32}$			
UNSIGNED48	$b_7..b_0$	$b_{15}..b_8$	$b_{23}..b_{16}$	$b_{31}..b_{24}$	$b_{39}..b_{32}$	$b_{47}..b_{40}$		
UNSIGNED56	$b_7..b_0$	$b_{15}..b_8$	$b_{23}..b_{16}$	$b_{31}..b_{24}$	$b_{39}..b_{32}$	$b_{47}..b_{40}$	$b_{55}..b_{48}$	
UNSIGNED64	$b_7..b_0$	$b_{15}..b_8$	$b_{23}..b_{16}$	$b_{31}..b_{24}$	$b_{39}..b_{32}$	$b_{47}..b_{40}$	$b_{55}..b_{48}$	$b_{63}..b_{56}$

3.3.1.2 Signed Integer

Daten vom Basis Datentyp INTEGERn definiert positive und negative Integer.
Der Wertebereich ist $-2^{n-1}-1..2^{n-1}-1$. Die Bitsequenz $b = b_0..b_{n-1}$ definiert den Wert

$$INTEGERn(b) = b_{n-2}\, 2^{n-2} +..+ b_1\, 2^1 + b_0\, 2^0 \text{ mit } b_{n-1} = 0$$

Negative Zahlen werden im zweier Komplement dargestellt - somit gilt:

$$INTEGERn(b) = -\, INTEGERn(b) - 1 \text{ mit } b_{n-1} = 1$$

Beispiel: Der Wert -266 = $FEF6_h$ wird mit dem Datentyp INTEGER16 mit Hilfe von zwei "octets" übertragen (1. octet = $F6_h$, 2. octet = FE_h).

Übertragungssyntax für den Datentyp INTEGERn:

Octet Nummer	1.	2.	3.	4.	5.	6.	7.	8.
INTEGER8	$b_7..b_0$							
INTEGER16	$b_7..b_0$	$b_{15}..b_8$						
INTEGER24	$b_7..b_0$	$b_{15}..b_8$	$b_{23}..b_{16}$					
INTEGER32	$b_7..b_0$	$b_{15}..b_8$	$b_{23}..b_{16}$	$b_{31}..b_{24}$				
INTEGER40	$b_7..b_0$	$b_{15}..b_8$	$b_{23}..b_{16}$	$b_{31}..b_{24}$	$b_{39}..b_{32}$			
INTEGER48	$b_7..b_0$	$b_{15}..b_8$	$b_{23}..b_{16}$	$b_{31}..b_{24}$	$b_{39}..b_{32}$	$b_{47}..b_{40}$		
INTEGER56	$b_7..b_0$	$b_{15}..b_8$	$b_{23}..b_{16}$	$b_{31}..b_{24}$	$b_{39}..b_{32}$	$b_{47}..b_{40}$	$b_{55}..b_{48}$	
INTEGER64	$b_7..b_0$	$b_{15}..b_8$	$b_{23}..b_{16}$	$b_{31}..b_{24}$	$b_{39}..b_{32}$	$b_{47}..b_{40}$	$b_{55}..b_{48}$	$b_{63}..b_{56}$

3.3.2 Zusammengesetzte Datentypen

Zusammengesetzte Datentypen nutzen Basisdatentypen (INTEGERn, UNSIGNEDn, REAL). Es werden zwei Datentypen unterschieden:

- STRUCT
  Dieser Datentyp setzt sich aus unterschiedlichen Basisdatentypen zusammen.
- ARRAY
  Dieser Datentyp setzt sich aus gleichen Basisdatentypen zusammen.

3.3.3 Erweiterte Datentypen

Erweiterte Datentypen werden aus Basisdatentypen und zusammengesetzten Datentypen abgeleitet. Im Folgenden werden die unterstützten Datentypen definiert.

3.3.3.1 Octet String

Der Datentyp "OCTET_STRING” definiert sich aus dem Datentyp ARRAY. "Length” ist die Länge des "octet string”.

ARRAY[length] OF UNSIGNED8	OCTET_STRINGlength

3.3.3.2 Visible String

Der Datentyp "VISIBLE_” kann sich aus dem Datentyp UNSIGNED8 oder aus dem Datentyp ARRAY definieren. Die zulässigen Werte sind 00_h und der Bereich von 20_h bis $7E_h$. Die Daten werden als 7 Bit ASCII Code interpretiert (ISO 646-1973(E)). "Length” ist die Länge des "visible string”.

UNSIGNED8	VISIBLE_CHAR
ARRAY[length] OF VISIBLE_CHAR	VISIBLE_STRINGlength

3.4 Kommunikationsobjekte

Die Kommunikationsobjekte werden mit Hilfe von Dienstelementen und Protokollen beschrieben. Es können im wesentlichen zwei Dienstelemente unterschieden werden:

- Unbestätigte Dienste PDO
- Bestätigte Dienste SDO

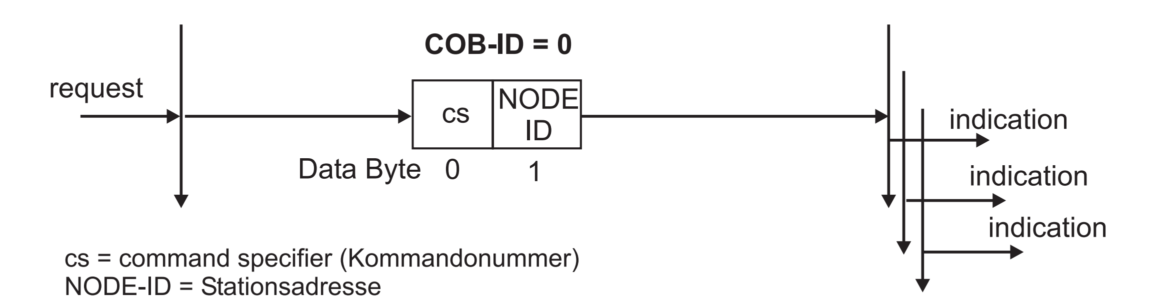
Sämtliche Dienste setzen voraus, das "Data Link” und "Physical Layer” fehlerfrei arbeiten.

Der S300/S700 unterstützt die Kommunikationsobjekte, die in den folgenden Kapiteln detailliert beschrieben werden:

- Network Management Objects (NMT)
- Synchronisation Object (SYNC)
- Emergency Object (EMCY)
- Process Data Object (PDO)
- Service Data Object (SDO)
- Nodeguard

3.4.1 Network Management Objects (NMT)

Das NMT - Telegramm sieht wie folgt aus:



Der Antrieb unterstützt folgende Funktionen des Netzwerkmanagements:

cs = 129, reset node: führt zu einem Kaltstart des Antriebs, damit werden sämtliche im RAM abgelegte Parameter gelöscht und wieder die im EEPROM gespeicherten Werte eingestellt.

cs = 130, reset communication node: stoppt die PDO Kommunikation, erzeugt eine neue Boot-up Meldung.

cs = 1, start remote node: startet den CAN-Knoten, d.h. gibt die PDOs des Antriebs zur Bedienung frei. Ab diesem Zeitpunkt werden Sende-PDOs ereignisgesteuert gesendet, bzw. kann der zyklische Prozessdatenbetrieb starten.

cs = 2, Stop remote node: stoppt den CAN-Knoten, d.h. der Antrieb reagiert nicht mehr auf empfangene PDOs und sendet keine mehr.

3.4.2 Synchronisation Object (SYNC)

Das SYNC Objekt wird meist als periodisches "Broadcast” – Objekt verwendet und gibt den Basis - Bustakt vor. Um einen zeitlich äquidistanten Abstand zu ermöglichen, besitzt das SYNC eine hohe Priorität. Die Anwendung dieses Protokolls wird im Anhang ab S.102 beschrieben. Mit Hilfe des SYNC-Objekt können z.B. auch Fahraufträge mehrerer Achsen gleichzeitig gestartet werden.

3.4.3 Time Stamp Object (TIME)

Dieses Kommunikationsobjekt wird von S300/S700 nicht unterstützt.

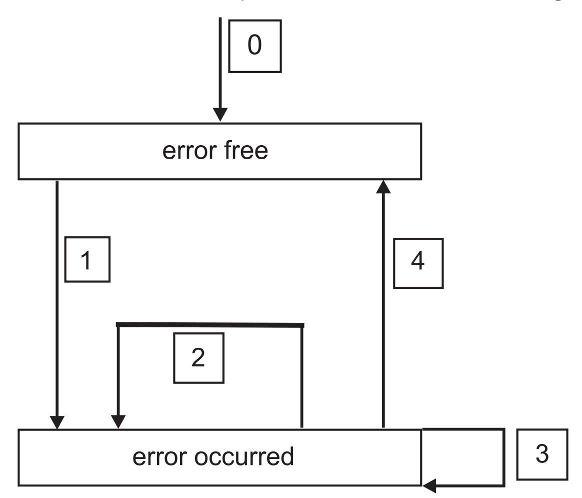
3.4.4 Emergency Object (EMCY)

Das EMCY wird ereignisgetriggert aufgrund einer internen Fehlersituation generiert. Für jeden Fehler wird dieses Objekt erneut übertragen. Da die Fehlercodes geräteabhängig sind, werden sie im Kapitel "CANopen Antriebsprofil" (⇒ S.25) beschrieben. Die letzten 8 Fehlercodes können über Objekt 1003 gelesen werden.

3.4.4.1 Verwendung des Emergency Object

Das Verhalten im Fehlerfall ist von der Fehlerart abhängig und somit unterschiedlich. Aus diesem Grund wird das Verhalten mit Hilfe eine Fehlerzustandsmaschine beschrieben. Es werden die Fehlerzustände "error free” und "error occurred” unterschieden.
Es sind die folgenden Übergänge definiert:

0. Nach der Initialisierung wird der Zustand "error free” eingenommen, falls kein Fehler erkannt wurde. In diesem Zustand wird keine Fehlermeldung generiert.
1. Der S300/S700 erkennt einen internen Fehler und zeigt diesen in den ersten drei Bytes des "emergency” Telegramms an ("error code” Byte 0,1 und "error register”Byte 2).
   Da der S300/S700 mehrere Fehlerarten unterscheidet, wird Byte 3 aus dem herstellerspezifischen Fehlerfeld dazu genutzt, um die Fehlerkategorie anzuzeigen.
2. Einer, aber nicht alle Fehler sind zurückgesetzt worden. Das EMCY Telegramm beinhaltet den "Error Code” 0000_h, das "Error Register” zeigt die restlichen noch anstehenden Fehler an. Der herstellerspezifische Bereich ist zu Null gesetzt.
3. Ein neuer Fehler ist aufgetreten. Der S300/S700 verbleibt im Zustand "Error State” und überträgt ein EMCY Objekt mit dem entsprechenden "Error Code”. Der neue "Error Code” wird in den Bytes 0 und 1 eingetragen.
4. Alle Fehler sind zurückgesetzt worden. Das EMCY Telegramm beinhaltet den "Error Code” 0000_h, das "Error Register” zeigt keine Fehler an.
   Der herstellerspezifische Bereich ist zu Null gesetzt.



3.4.4.2 Aufteilung des Emergency Object

Das Emergency - Objekt setzt sich aus 8 Byte zusammen und ist folgendermaßen aufgeteilt:

Byte	0	1	2	3	4	5	6	7
Inhalt	Emergency error code (⇒ S.25)		Error register (Objekt 1001_h)	Kategorie	Reserviert			

Wurde ein Emergency Objekt generiert, wird im Anschluss daran der Fehlerzustand der Zustandsmaschine gemeldet (error free / error occured), indem ein zweites Emergency - Objekt generiert wird. Hier sind nur die ersten 4 Bytes relevant (Emergency ErrorCode , Error register, Kategorie). Byte 0/1 enthält den "Error Reset Code" (0000_h) und Byte 2 zeigt an, dass eventuell noch ein Fehler ansteht. Ist das Error Register 00_h, ist der Fehlerzustand "error free".
Byte 3 enthält die Kategorie. Die Bedeutung der Fehlernummern (Error Code) und der Fehlerkategorie werden im Kapitel "Emergency Message” (⇒ S. 25) beschrieben. Das Fehlerregister wird über das Objekt 1001_h "Error register” definiert.

3.4.5 Service Data Objects (SDO)

Mit Hilfe der SDOs wird der Zugriff auf das Objektverzeichnis realisiert. Die SDOs werden zur Parametrierung und zur Statusabfrage verwendet. Der Zugriff auf ein einzelnes Objekt erfolgt mit einem Multiplexer über Index und Subindex des Objektverzeichnisses. Es werden die folgenden Kommunikationsprotokolle unterstützt:

- Initiate SDO Download Protocol
- Download SDO Segment Protocol
- Initiate SDO Upload Protocol
- Upload SDO Segment Protocol
- Abort SDO Transfer Protocol

Die Definition der einzelnen Kommunikationsdienste und der Protokolle sind dem DS301 zu entnehmen. Beispiele zur Handhabung von SDOs sind im Anhang ab S.102 zu finden.

INFO **Da es sich bei einem SDO um einen bestätigten Dienst handelt, muss immer auf das SDO - Antworttelegramm gewartet werden, bevor ein neues Telegramm gesendet werden darf.**

3.4.5.1 Aufbau eines Service Data Object

Ein SDO setzt sich aus folgenden Komponenten zusammen:

Byte	1	2	3	4	5	6	7	8
Inhalt	rw	Index		Subindex	Daten			

1. Das Kontrollbyte (Byte 1):
   Mit Hilfe des Kontrollbytes wird bestimmt, ob über das SDO schreibend oder lesend auf den Objektverzeichniseintrag zugegriffen wird.
   Eine Darstellung des gesamten Objektverzeichnis
   finden Sie ab Seite 97.
   Der Datenaustausch mit dem S300/S700 hält sich an den Standard "CMS multiplexed domain protocols", wie er im CAN-Standard DS 202 beschrieben wird.
   Um Daten zu lesen, muss das Kontrollbyte entsprechend folgender Darstellung beschrieben werden:

Bit	7	6	5	4	3	2	1	0
Inhalt	ccs*=2			X	X	X	X	X

* ccs ⇒ client command specifier (ccs = 2 ⇒ initiate upload request)
X ⇒ beliebig

Der Wert 0100 0000 (binary) oder 40h muss im Control-Byte übertragen werden.
Der Servoverstärker antwortet entsprechend mit einem Antwortbyte:

Bit	7	6	5	4	3	2	1	0
Inhalt	scs*=2			X	n		e	s

* scs ⇒ server command specifier (scs = 2 ⇒ initiate upload response)
n ⇒ nur gültig bei e = s = 1, Ist dies der Fall, so steht in n die Zahl der Bytes, die keine Daten enthalten.
X ⇒ beliebig

Wenn der Lesezugriff erfolgreich war, sind im Antwortbyte immer die Bits 0 und 1 auf 1 gesetzt (e=s=1)
Kodierte Byte-Länge in der SDO Antwort::
0x43 - 4 Bytes
0x47 - 3 Bytes
0x4B - 2 Bytes
0x4F - 1 Byte.

Wenn ein Fehler auftritt, wird scs auf 4 gesetzt, das Antwort-Byte ist 0x80 und die Fehlerinformation ist im 4-Byte Datenfeld enthalten. Aufschlüsselung des Fehlers: S. 21.

Um Daten zu schreiben, muss das Kontrollbyte entsprechend folgender Darstellung beschrieben werden:

Client	*Initiate Domain Download*															Server
	Byte 1								2	3	4	5	6	7	8	
request	7	6	5	4	3	2	1	0								indication
⇒	ccs=1			X	n		e	s	m			d				⇒

⇒⇒⇒⇒⇒⇒⇒⇒⇒⇒⇒⇒⇒⇒⇒⇒⇒⇒⇒⇒⇒⇒⇒⇒⇒⇒⇒⇒⇒⇒⇒

n,e und s sind wie im Lesefall definiert., m: Index + Subindex, d: Datenfeld mit 4 Bytes
Die Datenlänge eines Objekts finden Sie im Objekt-Verzeichnis im Anhang.
Das Control Byte soll sein:
0x23 für 4-Byte Zugriff
0x27 für 3-Byte Zugriff
0x2B für 2-Byte Zugriff
0x2F für 1-Byte Zugriff

⇐⇐⇐⇐⇐⇐⇐⇐⇐⇐⇐⇐⇐⇐⇐⇐⇐⇐⇐⇐⇐⇐⇐⇐⇐⇐⇐⇐⇐⇐⇐

⇐	Byte 1								2	3	4	5	6	7	8	⇐
confirm	7	6	5	4	3	2	1	0								response
	scs=3			X					m			reserved				

2. Der Index (Bytes 2 und 3):
   Der Index ist der Haupteintrag im Objektverzeichnis und teilt die Parameter in Gruppen ein (Beispiel: Index 1018_h – Identitäts Objekt). Wie bei allen CAN – Daten wird der Index in umgekehrter Reihenfolge in den Bytes abgelegt,
   z.B.: Index 6040_h -> Byte 2 = 40_h, Byte 3 = 60_h)

3. Der Subindex (Byte 4):
   Der Subindex unterteilt die Parameter innerhalb einer Parametergruppe.

4. Datenfeld (Bytes 5 – 8):
   Über diese Komponenten werden die eigentlichen Nutzdaten ausgetauscht. In den Telegrammen zur Leseaufforderung an den S300/S700 sind diese auf 0 zu setzen, in der Schreibbestätigung vom S300/S700 haben diese Daten bei erfolgreichem Transfer keinen Inhalt, bei fehlerhaftem Schreiben enthalten sie einen Fehlercode (⇒ 3.4.5.6).

3.4.5.2 Initiate SDO Download Protocol

Das "Initiate SDO Download" – Protokoll wird für Schreibzugriffe auf Objekte mit bis zu 4 Byte Nutzdaten verwendet ("expedited transfer") oder zum Einleiten eines Segmenttransfers ("normal transfer").

Der "expedited transfer" wird auch für Objekte genutzt, die nur eine Kommandoeigenschaft aufweisen (z.B. ASCII: SAVE) und somit keine weiteren Nutzdaten benötigen.

3.4.5.3 Download SDO Segment Protocol

Das "Download SDO Segment" – Protokoll wird für Schreibzugriffe auf Objekte mit mehr als 4 Byte Nutzdaten verwendet ("normal transfer"). Dieser Dienst wird vom S300/S700 z.Zt. nicht unterstützt, da keine Objekte existieren, die mehr als 4 Byte Nutzdaten verarbeiten.

3.4.5.4 Initiate SDO Upload Protocol

Das "Initiate SDO Upload" – Protokoll wird für Lesezugriffe auf Objekte mit bis zu 4 Byte Nutzdaten ("expedited transfer") oder zum Einleiten eines Segmenttransfers ("normal transfer") verwendet .

3.4.5.5 Upload SDO Segment Protocol

Das "Upload SDO Segment" – Protokoll wird für Lesezugriffe auf Objekte mit mehr als 4 Byte Nutzdaten verwendet ("normal transfer"). Dieser Dienst wird vom S300/S700 z.Zt. nicht unterstützt, da keine Objekte existieren, die mehr als 4 Byte Nutzdaten verarbeiten.

3.4.5.6 Abort SDO Protocol

Das "Abort SDO" Protokoll wird bei einer fehlerhaften SDO-Übertragung vom S300/S700 verwendet und zeigt mit Hilfe des "abort code" (Fehlercode) den Fehler an, der zum Abbruch der Übertragung führte. Der Fehlercode ist als UNSIGNED32-Wert kodiert. Die folgende Tabelle zeigt die möglichen Ursachen für einen Abbruch:

Abort Code	Beschreibung
0601 0000$_h$	Zugriff auf dieses Objekt wird nicht unterstützt
0601 0001$_h$	Versuchter Lesezugriff auf ein Objekt, das nur geschrieben werden kann
0601 0002$_h$	Versuchter Schreibzugriff auf ein Objekt, das nur gelesen werden kann
0602 0000$_h$	Objekt existiert im Objektverzeichnis nicht
0604 0041$_h$	Objekt kann nicht in ein PDO "gemappt" werden
0604 0042$_h$	Größe und Anzahl der "gemappten" Objekte übersteigt die mögliche PDO Länge
0604 0043$_h$	Allgemeine Parameter - Inkompatibilität
0607 0010$_h$	Datentyp passt nicht, Länge des Service Parameters passt nicht
0609 0011$_h$	Subindex existiert nicht
0609 0030$_h$	Wertebereich des Parameters verlassen (nur für Schreibzugriffe)
0609 0031$_h$	Wert des Parameters zu groß
0609 0032$_h$	Wert des Parameters zu klein
0800 0020$_h$	Datum kann nicht übertragen oder gespeichert werden
0800 0022$_h$	Datum kann aufgrund des Gerätezustandes nicht übertragen / gespeichert werden
FF03 0000$_h$	OS Kommandospeicher voll

Nicht aufgeführte "Abort Codes" sind reserviert.

3.4.6 Process Data Object (PDO)

PDOs werden für die Echtzeit - Datenkommunikation verwendet. Zum Beispiel lassen sich über PDOs ähnliche Regler aufbauen wie sie bei analog realisierten Antrieben verwendet werden. Statt der dort verwendeten +/- 10V Sollwertvorgabe und ROD-Istwertausgabe können hier digitaler Drehzahlsollwert und Istpositionsausgabe über PDOs realisiert werden. Die PDO-Daten werden dazu ohne Protokoll-Overhead übertragen und der Empfang nicht bestätigt.
Dieses Kommunikationsobjekt nutzt einen unbestätigten Kommunikationsdienst.
PDOs definieren sich über das Objektverzeichnis des S300/S700. Das "Mapping" erfolgt mit Hilfe von SDOs während der Konfigurationsphase. Die Länge wird über die gemappten Objekte definiert.

Die Definition des PDO Dienstes und des Protokolls sind dem DS301 zu entnehmen. Beispiele zur Handhabung von PDOs sind im Anhang ab S.102 zu finden.
Grundsätzlich können zwei Arten von PDOs in Abhängigkeit der Übertragungsrichtung unterschieden werden:

- Transmit PDOs (TPDOs) (S300/S700 ⇒ Master)
  Die TPDOs übertragen Daten vom S300/S700 zur Steuerung
  (z.B. Istwertobjekte, Gerätezustände).
- Receive PDOs (RPDOs) (Master ⇒ S300/S700)
  Die RPDOs übertragen Daten von der Steuerung zum S300/S700
  (z.B. Sollwerte)

Der S300/S700 unterstützt für jede Übertragungsrichtung vier unabhängige PDO - Kanäle. Die Kanäle sind mit den Kanalnummern 1 bis 4 gekennzeichnet.
Zur Konfiguration der vier möglichen PDOs stehen jeweils zwei Parametersätze zur Verfügung, die über entsprechende SDOs eingestellt werden können:

1. Mappingparameter, über die ermittelt werden kann, welche Daten im ausgewählten PDO vorhanden (= gemappt) sind und über die bestimmt werden kann, welche Daten enthalten sein sollen (siehe Seiten 51 und 53).
2. Kommunikationsparameter, die festlegen, ob die PDOs im synchronisierten Betrieb oder ereignisgesteuert arbeiten sollen: (Objekte 1400$_h$ bis 1403$_h$, 1800$_h$ bis 1803$_h$)

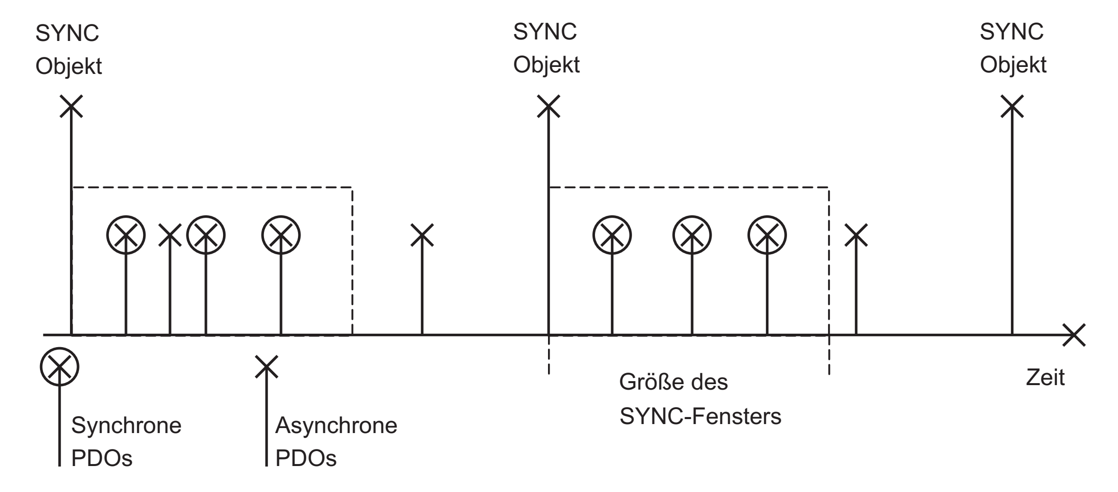
3.4.6.1 Übertragungsmodi

Es werden die folgenden PDO Übertragungsmodi (transmission mode) unterschieden:

- Synchrone Übertragung
- Asynchrone Übertragung

Um Antriebe zu synchronisieren, wird periodisch das vordefinierte SYNC – Objekt übertragen (Bustakt). Synchrone PDOs werden innerhalb eines vordefinierten Zeitfensters im Anschluss an das SYNC Objekt übertragen.

Die Einstellung der Übertragungsmodi wird mit Hilfe der PDO Kommunikationsparameter vorgenommen.



3.4.6.2 Triggermodi

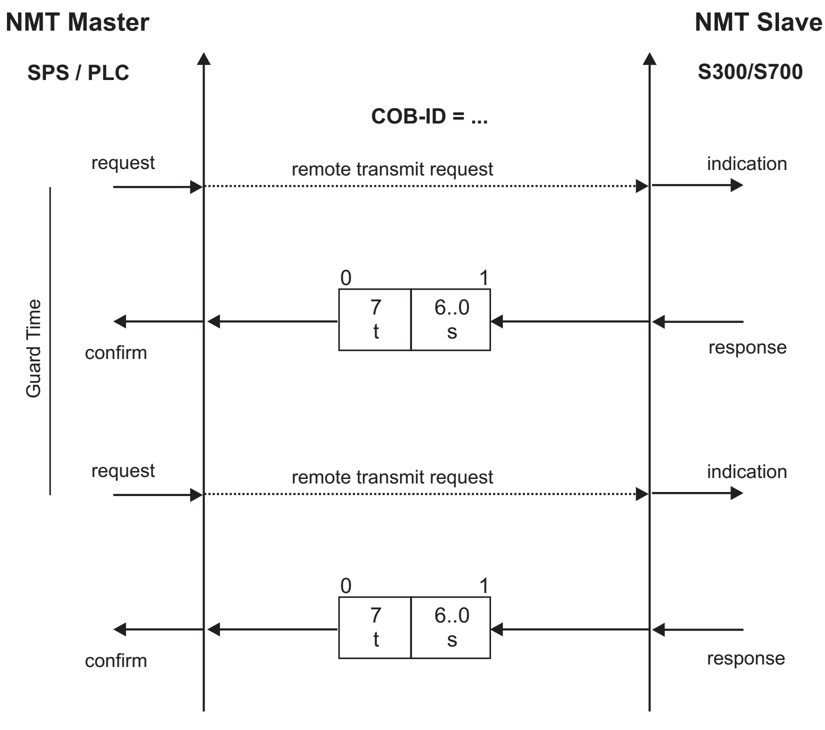
Es werden drei Triggermodi (Triggering Modes) unterschieden:

- **Event Driven**
  Die Übertragung der Telegramme wird über ein objektspezifisches Ereignis getriggert.
- **Time Driven**
  Bei ereignisgesteuerten Signalen, die eine zu hohe Buslast erzeugen würden, kann mit Hilfe der "inhibit time" (Kommunikationsparameter, Subindex 03h) betimmt werden, nach welcher Mindestzeit wieder ein PDO gesendet werden darf.
- **Event Timer driven**
  Wenn ein PDO (auch ohne Änderung) während eines bestimmten Zeitintervalls gesendet werden soll, kann dieses Intervall mit einem speziellen SDO definiert werden.

3.4.7 Nodeguard

Das Node Guarding Protokoll dient der Funktionsüberwachung des Antriebs. Dazu wird der Antrieb in äquidistanten Zeitabständen vom CANopen - Master angesprochen.
Der maximale zeitliche Abstand, der zwischen zwei Nodeguard - Telegrammen liegen darf, wird durch das Produkt der Guard Time (Objekt $100C_h$, ⇒ S.30) und des Life Time Factors (Objekt $100D_h$, ⇒ S.30) bestimmt. Ist einer dieser beiden Werte 0, ist die Ansprechüberwachung deaktiviert. Das Wird der Antrieb innerhalb der Zeit, die durch die Objekts $100C_h$ und $100D_h$ definiert ist, nicht angesprochen, tritt die Warnung N04 (Ansprechüberwachung) am Antrieb auf, der Antrieb wird mittels Quickstop - Rampe abgebremst und jede weitere Bewegung wird verhindert (Parameter DECSTOP, Objekt 6085 sub0).

Die zeitliche Folge des Nodeguardings sieht wie folgt aus:

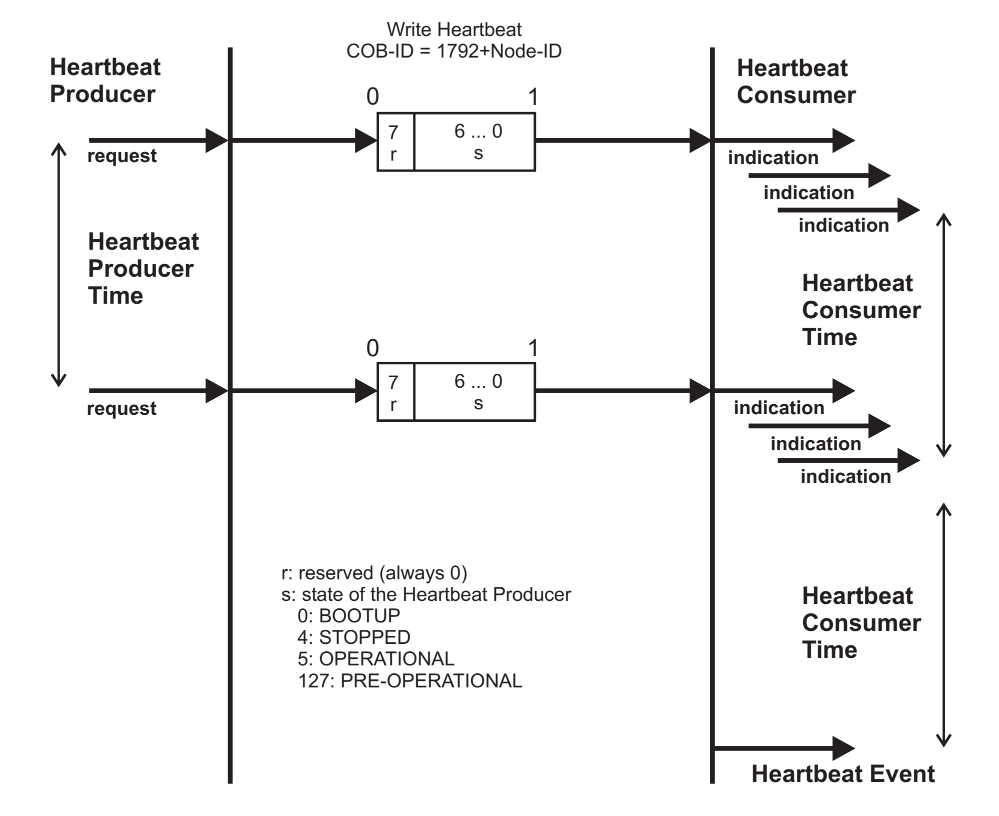


t = toggle Bit, ändert seinen Zustand mit jedem Slave-Telegramm
s = Zustand der NMT Slave - Zustandsmaschine
Das Nodeguarding wird vom Master mit RTR - Telegrammen mit dem COB-ID 700_h + Slave - Knotenadresse ausgeführt.

3.4.8 Heartbeat

Das Heartbeat Protokoll definiert einen Fehlerüberwachungsdienst ohne Remote Frame. Ein Heartbeat Producer überträgt zyklisch eine Heartbeat Nachricht. Einer oder mehrere Heartbeat Consumer empfangen die Meldung. Die Beziehung zwischen Producer und Consumer ist mit Hilfe von Objekt 1016h / 1017h konfigurierbar. Der Heartbeat Consumer überwacht den Empfang des Heartbeat während der Heartbeat Consumer Time. Wenn der Heartbeat während dieser Zeit nicht empfangen wird, wird ein Heartbeat Event erzeugt.

Heartbeat Protokoll:



4 CANopen Antriebsprofil

4.1 Emergency Messages

"Emergency Messages" werden durch interne Gerätefehler ausgelöst. Sie haben eine hohe ID – Priorität, um einen schnellen Buszugriff sicherzustellen. Die "Emergency Message" beinhaltet ein Fehlerfeld mit vordefinierten Fehlernummern (2 Byte), Fehlerregister (1Byte) und die Fehlerkategorie (1Byte) und zusätzlichen Informationen (⇒ Kap. 3). Das höherwertige Byte der Fehlernummer gibt die Fehlerklasse und das niederwertige Byte die Fehlernummer der Klasse an.
Fehlernummern von 0000_h bis $7FFF_h$ sind im Kommunikations- oder Antriebsprofil definiert. Fehlernummern von $FF00_h$ bis $FFFF_h$ sind herstellerspezifische Definitionen. Mit Hilfe der Fehlerkategorie können auftretende Fehler in ihrer Bedeutung klassifiziert werden. Es sind die folgenden Fehlerkategorien definiert:

1: Fehler, die nur durch einen Reset (Kommando "COLDSTART" oder Bit 7 im controlword ⇒ S.57) gelöscht werden können. Wird ein Reset bei einem Fehler dieser Kategorie ausgeführt, wird ein Coldstart des Geräts ausgeführt.
2: Fehler, die durch Bit 7 im controlword (⇒ S. 57) gelöscht werden können.
3: Fehlermeldungen, die bei der Verarbeitung eines PDOs auftreten können.
4: Fehler, die **nicht** durch den Anwender behoben werden können.
5: Bedienungsfehler / Warnungen

Die Fehler der Kategorien 1 und 2 werden durch Blinken der LED-Anzeige in der Frontplatte angezeigt (Fxx, xx = Fehlernummer). Die folgende Tabelle beschreibt die verschiedenen Error Codes:

Error Code	Kategorie	Beschreibung
0000_h	—	Error reset or no error (mandatory)
1000_h	—	Generic error (mandatory)
1080_h	5	Kein BTB vorhanden (Zustand "not ready for operation")
2330h	2	Fehler im Erdanschluss (F22)
2380h	1	Fehler im Motor Anschluss (Phasenfehler) (F12)
3100_h	2	Kein Netz-BTB (F16)
3110_h	2	Überspannung Zwischenkreis (F02)
3120_h	2	Unterspannung Zwischenkreis (F05)
3130_h	2	Eine Netzphase fehlt (bei PMODE = 2) (F19)
4110_h	2	Umgebungstemperatur überschritten (F13)
4210_h	1	Kühlkörpertemperatur überschritten (F01)
4310_h	1	Motortemperatur überschritten (F06)
5111_h	1	Fehler ±15VHilfsspannung (F07)
5380_h	1	Fehler A/D-Konverter (F17)
5400_h	1	Endstufenfehler (F14)
5420_h	1	Ballast (Chopper) (F18)
5441_h	1	Fehler bei Bedienung der AS - Option (F27)
5530_h	1	Serielles EEPROM (F09)
6320_h	3	Parameterfehler
7111_h	1	Bremsenfehler (F11)
7122_h	1	Kommutierungsfehler (F25)
7181_h	5	S300/S700 konnte nicht enabled werden
7303_h	1	Fehler Rückführeinheit (F04)
7305h	1	Signalfehler digitaler Encoder Eingang (F10)
8182_h	1	CAN - Busoff (F23)
8331_h	2	I^2t (torque fault, F15)
8480_h	2	Überdrehzahl (overspeed, F08)
8611_h	2	Schleppfehler (n03/F03)
8681_h	5	Ungültige Fahrauftragsnummer
$FF01_h$	1	Schwerwiegender Ausnahmefehler (F32)
$FF02_h$	3	Fehler in PDO-Komponente
$FF04_h$	1	Slotfehler (F20)
$FF05_h$	1	Handlingfehler (F21)
$FF06_h$	2	Warnungsanzeige als Fehler (F24)
$FF07_h$	2	Fehler Referenzfahrt (HW-Endschalter erreicht) (F26)
$FF08_h$	2	Sercos - Fehler (F29)
$FF11_h$	2	Emergency Timeout Ausfall (F30)

4.2 Allgemeine Definitionen

In diesem Kapitel werden allgemeingültige Objekte beschrieben (z.B. Objekt 1000_h "Device Type"). Im Anschluss daran wird die freie Konfiguration von Prozessdatenobjekten ("free mapping") erläutert.

4.2.1 Allgemeine Objekte

4.2.1.1 Objekt 1000h: Device Type (DS301)

Dieses Objekt beschreibt den Gerätetyp (Servoantrieb) und die Gerätefunktionalität (DS402 Antriebsprofil). Es setzt sich folgendermaßen zusammen:

MSB LSB

zusätzliche Informationen		Geräte-Profil-Nummer
Modebits	Typ	402_d=192_h
31 ... 24	23 ... 16	15 ... 0

Die Geräteprofilnummer ist DS402, der Typ ist 2 für Servoverstärker, die Betriebsart-Bits 28 bis 31 sind herstellerspezifisch und können von aktuellen Wert auf 0 geändert werden. Lesezugriff liefert zur Zeit 0x00002192.

Objektbeschreibung:

Index	**1000_h**
Name	Gerätetyp
Objektcode	VAR
Datentyp	UNSIGNED32
Kategorie	mandatory

Wertbeschreibung:

Zugriff	ro
PDO Mapping	nicht möglich
Wertebereich	UNSIGNED32
Defaultwert	—

4.2.1.2 Objekt 1001h: Error register (DS301)

Fehlerregister für das Gerät. Das Gerät kann interne Fehler in dieses Register eintragen. Dieses Objekt ist Teil des Fehlerobjekts (Emergency Message).

Index	**1001_h**
Name	Error register
Objektcode	VAR
Datentyp	UNSIGNED8
Kategorie	mandatory

Zugriff	ro
PDO Mapping	nicht möglich
Wertebereich	UNSIGNED8
Defaultwert	—

Im Folgenden wird die Bitbelegung des Fehlerregisters beschrieben. Ein gesetztes Bit zeigt ein Fehlerereignis an. Bit 0 wird in jedem Fehlerfall gesetzt.

Bit	Beschreibung	Bit	Beschreibung
0	generic error	4	communication error (Überlauf, Fehlerstatus)
1	current	5	device profile specific
2	voltage	6	reserved (immer 0)
3	temperature	7	manufacturer specific

4.2.1.3 Objekt 1002h: Manufacturer Status Register (DS301)

Das herstellerspezifische Statusregister enthält wichtige Verstärker-Informationen.

Index	**1002$_h$**
Name	Manufacturer Status Register
Objektcode	VAR
Datentyp	UNSIGNED32
Kategorie	optional

Zugriff	ro
PDO Mapping	möglich
Wertebereich	UNSIGNED32
Defaultwert	—

Die folgende Tabelle zeigt die Bitbelegung des Statusregisters:

Bit	Beschreibung
0	1 = Bewegung aktiv (Positionierung, Referenzfahrt)
1	1 = Referenzpunkt gesetzt
2	1 = Referenzschalter betätigt (Home Position)
3	1 = In Position
4	1 = Position Latch an Eingang 2 (positiver Übergang)
5	reserviert
6	reserviert
7	reserviert
8	reserviert
9	reserviert
10	1 = Initialisierungsphase beendet
11	reserviert
12	1 = Motorstillstands-Meldung (threshold VEL0)
13	1 = Sicherheitsrelais angewählt (AS)
14	1 = Endstufe freigegeben
15	1 = Fehlerstatus
16	1 = Referenzfahrt aktiv
17	1 = Tippbetrieb aktiv
18	1 = Position Latch an Eingang 2 (negativer Übergang)
19	1 = Notstopp aktiv
20	1 = Position Latch an Eingang 1 (positiver Übergang)
21	1 = Position Latch an Eingang 1 (negativer Übergang)
22	1 = Feed forward ausgeschaltet
23	1 = Referenzfahrt beendet
24	1 = ein aktueller Fehler wird beim Reset einen Kaltstart auslösen
25	1 = Digitaler Eingang 1 gesetzt
26	1 = Digitaler Eingang 2 gesetzt
27	1 = Digitaler Eingang 3 gesetzt
28	1 = Digitaler Eingang 4 gesetzt
29	1 = Digitaler Eingang Hardware Enable gesetzt
30	reserviert
31	reserviert

4.2.1.4 Objekt 1003h: Pre-Defined Error Field (DS301)

Objekt 1003h liefert eine Fehlerhistorie mit maximal 8 Einträgen.
Subindex 0 beinhaltet die Anzahl der aufgetretenen Fehler seit dem letzten Reset der Fehlerhistorie (entweder beim Start des Verstärkers oder durch Schreiben einer 0 in Subindex 0).

Eine neue Fehlermeldung wird in Subindex 1 geschrieben, der alte Eintrag wird dabei in den nächsthöheren Index geschoben. Der Inhalt von Subindex 8 geht verloren.

Die UNSIGNED32 Information, die in die Sub-Indizes geschrieben wird, ist im Felder Error Code in der Tabelle bei der Beschreibung der Emergency Messages (⇒ S.25) definiert.

Index	**1003_h**
Name	predefined Error Field
Objektcode	ARRAY
Datentyp	UNSIGNED32
Kategorie	optional

Subindex	**0**
Beschreibung	Anzahl der Einträge
Datentyp	UNSIGNED8
Kategorie	mandatory
Zugriff	rw
PDO Mapping	nicht möglich
Wertebereich	0...8
Defaultwert	0

Subindex	**1 ... 8**
Beschreibung	Standard Fehlerfeld (⇒ S. 25)
Kategorie	optional
Zugriff	ro
PDO Mapping	nicht möglich
Wertebereich	UNSIGNED32
Defaultwert	—

4.2.1.5 Objekt 1005h: COB-ID der SYNC - Message (DS301)

Dieses Objekt definiert die COB-ID des SYNC-Objekts.

Index	**1005_h**
Name	COB - ID der SYNC-Mssage
Objektcode	VAR
Datentyp	UNSIGNED32
Kategorie	conditional

Zugriff	rw
PDO Mapping	nicht möglich
Wertebereich	UNSIGNED32
Defaultwert	—

Bit kodierte Information:

Bit	Wert	Bedeutung
31 (MSB)	X	—
30	0	Gerät erzeugt keine SYNC-Message
	1	Gerät erzeugt SYNC-Message
29	0	11 Bit ID (CAN 2.0A)
	1	29 Bit ID (CAN 2.0B)
28..11	X	wenn Bit 29=1 => Bit 11 ... 28 der 29-bit SYNC COB-ID
	0	wenn Bit 29=0
10 ... 0 (LSB)	X	Bit 0..10 von SYNC COB-ID

4.2.1.6 Objekt 1006h: Communication Cycle Period (DS301)

Mit diesem Objekt kann die Periode des Sendens des SYNC - Telegramms in µs festgelegt werden.

Index	**1006_h**
Name	Periode des Kommunikationszyklus
Objektcode	VAR
Datentyp	UNSIGNED32
Kategorie	optional

Zugriff	rw
PDO Mapping	nicht möglich
Wertebereich	UNSIGNED32
Defaultwert	00_h

4.2.1.7 Objekt 1008h: Manufacturer Device Name (DS301)

Der Gerätename besteht aus vier ASCII-Zeichen und hat die Buchstaben "S3xx” zum Inhalt, wobei xx für die Stromstärke der Endstufe steht.

Index	**1008_h**
Name	Manufacturer Device Name
Objektcode	VAR
Datentyp	Visible String
Kategorie	Optional

Zugriff	const
PDO Mapping	nicht möglich
Wertebereich	S301 - S3xx
Defaultwert	—

4.2.1.8 Objekt 1009h: Manufacturer Hardware Version

Die Hardware Version des Servoverstärkers wird angezeigt.

Index	**1009_h**
Name	manufacturer hardware version
Objektcode	VAR
Datentyp	Visible String
Kategorie	Optional

Zugriff	const
PDO Mapping	nicht möglich
Wertebereich	—
Defaultwert	—

4.2.1.9 Objekt 100Ah: Manufacturer Software Version (DS301)

Das Objekt beinhaltet die Version der Herstellersoftware (hier: CANopen Teil der Verstärker- Firmware)

Index	**$100A_h$**
Name	Manufacturer Software Version
Objektcode	VAR
Datentyp	Visible String
Kategorie	Optional

Zugriff	const
PDO Mapping	nicht möglich
Wertebereich	0.01 ... 9.99
Defaultwert	—

4.2.1.10 Objekt 100Ch: Guard Time (DS301)

Das Produkt der Objekte 100C$_h$ "Guard Time" und 100D$_h$ "Life Time Factor" ergibt die Ansprechüberwachungszeit. Die "Guard Time" wird in Millisekunden angegeben. Die Ansprechüberwachung wird erst mit dem ersten "Nodeguard - Objekt" aktiv (⇒ S. 23). Wird der Wert des Objekts "Guard Time" zu Null gesetzt, ist die Ansprechüberwachung inaktiv.

Index	**100C$_h$**
Name	Guard Time
Objektcode	VAR
Datentyp	UNSIGNED16
Kategorie	conditional; mandatory, wenn Heartbeat nicht unterstützt wird

Zugriff	rw
PDO Mapping	nicht möglich
Wertebereich	UNSIGNED16
Defaultwert	0

4.2.1.11 Objekt 100Dh: Life Time Factor (DS301)

Das Produkt aus Guard Time und Life Time Faktor ergibt die Lebenszeit für das NodeGuarding Protokoll. Wenn dies 0 ergibt, wird das Protokoll nicht verwendet.

Index	**100D$_h$**
Name	Life Time Factor
Objektcode	VAR
Datentyp	UNSIGNED8
Kategorie	conditional; mandatory, wenn Heartbeat nicht unterstützt wird

Zugriff	rw
PDO Mapping	nicht möglich
Wertebereich	UNSIGNED8
Defaultwert	0

4.2.1.12 Objekt 1010h: Store Parameters (DS301)

Mit Hilfe dieses Objekts können die Parameter in einem Flash-EEProm im Servobverstärker abgespeichert werden. Nur Subindex 1 wird unterstützt (alle Parameter sichern, ähnlich wie das Sichern als Datei mit Hilfe der Inbetriebnahmesoftware).

Index	**1010_h**
Name	store parameters
Objektcode	ARRAY
Datentyp	UNSIGNED32
Kategorie	optional

Subindex	**0**
Name	Anzahl der Einträge
Object code	VAR
Data type	UNSIGNED8
Category	mandatory
Access	ro
PDO Mapping	nicht möglich
Value range	1
Default value	1

Subindex	**1**
Name	Sichere alle Parameter
Object code	VAR
Data type	UNSIGNED32
Category	mandatory
Access	rw
PDO Mapping	nicht möglich
Value range	UNSIGNED32
Default value	1

Datendefinition:

Bit Nummer	Wert	Bedeutung
31 ... 2	0	reserviert (=0)
1	0	Gerät speichert Parameter nicht selbständig
	1	Gerät speichert Parameter selbständig
0	0	Gerät speichert Parameter nicht auf Befehl
	1	Gerät speichert Parameter auf Befehl

Beim Lesezugriff aus Subindex 1 liefert der Servoverstärker Informationen über seine Speicherfunktionalität.

Der S300/S700 liefert einen konstanten Wert 1 beim Lesezugriff. Das bedeutet, alle Parameter können durch Schreiben auf Objekt1010 sub 1 gespeichert werden. Normalerweise speichert der Verstärker die Parameter nicht selbständig, spezielle Ausnahmesituation ist z.B. Homing von Multiturn Absolut-Encodern.

Die Parameter werden nur gespeichert, wenn eine spezielle Zeichenkette ("save") in Subindex 1 geschrieben wird. "save" entspricht der Unsigned32-Zahl 65766173_h.

4.2.1.13 Objekt 1011h: Restore default parameters

Mit diesem Objekt werden die Defaultwerte der Parameter bezogen auf die Kommunikation oder das Geräteprofil wiederhergestellt. Der S300/S700 ermöglicht die Wiederherstellung aller Defaultwerte.

Index	**1011$_h$**
Name	restore default parameters
Objektcode	ARRAY
Datentyp	UNSIGNED32
Kategorie	optional

Subindex	**0**
Name	Anzahl der Einträge
Datentyp	UNSIGNED8
Kategorie	mandatory
Zugrif	ro
PDO Mapping	nicht möglich
Defaultwert	1

Subindex	**1**
Name	Stelle alle Default Parameter wieder her
Kategorie	mandatory
Zugriff	rw
PDO Mapping	nicht möglich
Wertebereich	UNSIGNED32 (Figure 57)
Defaultwert	1 (Gerät stellt Parameter wieder her)

Die Wiederherstellung wird nur ausgeführt, wenn eine spezielle Signatur ("load") in Subindex 1 geschrieben wird. "load" muss als Unsigned32 - Zahl 64616F6C$_h$ übertragen werden.

4.2.1.14 Objekt 1014h: COB - ID für Emergency Message (DS301)

Dieses Objekt definiert die COB - ID der Emergency Message.

Index	**1014$_h$**
Name	COB-ID emergency message
Objektcode	VAR
Datentyp	UNSIGNED32
Kategorie	conditional; mandatory, wenn Emergency unterstützt wird

Zugriff	rw
PDO Mapping	nicht möglich
Wertebereich	UNSIGNED32
Defaultwert	80$_h$ + Node - ID

4.2.1.15 Objekt 1016h: Consumer Heartbeat Time

Die Consumer Heartbeat Time definiert die erwartete Heartbeat-Zykluszeit und muss größer sein als die zugehörige "Producer Heartbeat Time", die auf dem Gerät konfiguriert wurde, die sie erzeugt.

Die Überwachung startet nach Empfang des ersten Heartbeat. Ist die Consumer Heartbeat Time gleich 0, wird der Eintrag nicht benutzt. Die Zeit ist definiert in ms.

Index	**1016_h**
Name	consumer heartbeat time
Objektcode	ARRAY
Datentyp	UNSIGNED32
Kategorie	optional

Subindex	**0**
Beschreibung	Anzahl der Einträge
Datentyp	UNSIGNED8
Kategorie	mandatory
Zugriff	ro
PDO Mapping	nicht möglich
Wertebereich	1
Defaultwert	1

Subindex	**1**
Beschreibung	Consumer heartbeat time
Kategorie	mandatory
Zugriff	rw
PDO Mapping	nicht möglich
Wertebereich	UNSIGNED32
Defaultwert	—

Definition Werte in Subindex 1

	MSB		LSB
Wert	reserviert (value: 00)	Node-ID	heartbeat time
Kodiert als	-	UNSIGNED8	UNSIGNED16
Bit	31 24	23 16	15 0

4.2.1.16 Objekt 1017h: Producer Heartbeat Time

Die Producer Heartbeat Time definiert die Zykluszeit des Heartbeat in ms. Bei 0 wird sie nicht verwendet.

Index	**1017_h**
Name	Producer heartbeat time
Objektcode	VAR
Datentyp	UNSIGNED16
Kategorie	conditional; mandatory, wenn Guarding nicht unterstützt wird

Zugriff	rw
PDO Mapping	nicht möglich
Wertebereich	UNSIGNED16
Defaultwert	0

4.2.1.17 Objekt 1018h: Identity Object (DS301)

Das Identitätsobjekt beinhaltet allgemeine Geräteinformationen.

Index	**1018_h**
Name	Identity Object
Objektcode	RECORD
Datentyp	Identity
Kategorie	mandatory

Subindex	**0**
Beschreibung	Anzahl der Einträge
Datentyp	UNSIGNED8
Kategorie	mandatory
Zugriff	ro
PDO Mapping	nicht möglich
Wertebereich	1 ... 4
Defaultwert	4

Subindex 1 ist eine einzigartige Nummer für einen Gerätehersteller.

Subindex	**1**
Beschreibung	Vendor ID
Kategorie	mandatory
Zugriff	ro
PDO Mapping	nicht möglich
Wertebereich	UNSIGNED32
Defaultwert	$0x6A_H$ (Kollmorgen)

Subindex 2 enthält die allgemeine Gerätenummer (300) und Informationen über DC-Bus Spannung und Stromklasse.

Subindex	**2**
Beschreibung	Product Code
Kategorie	optional
Zugriff	ro
PDO Mapping	nicht möglich
Wertebereich	301 ... 346
Defaultwert	—

Subindex 3 besteht aus zwei Revisionsnummern:
- die Haupt-Revisionsnummer im oberen Wort enthält die CAN-Version
- die Neben-Revisionsnummer enthält die Firmware-Version
Z.B.: bedeutet der Wert 0x0022 0079: CAN-Version 0.34 und Firmware-Version 1.21.

Subindex	**3**
Beschreibung	Revision Number
Kategorie	optional
Zugriff	ro
PDO Mapping	nicht möglich
Wertebereich	UNSIGNED32
Defaultwert	—

Subindex 4 enthält die Seriennummer des Gerätes..

Subindex	**4**
Beschreibung	Serial Number
Kategorie	optional
Zugriff	ro
PDO Mapping	nicht möglich
Wertebereich	UNSIGNED32
Defaultwert	—

4.2.1.18 Objekt 1026h: OS Prompt

Der OS Prompt wird für den Aufbau eines ASCII Kommunikationskanals zum Verstärker benutzt.

Index	**1026$_h$**
Name	OS Prompt
Objektcode	ARRAY
Datentyp	UNSIGNED8
Kategorie	optional

Subindex	**0**
Beschreibung	Anzahl der Einträge
Datentyp	UNSIGNED8
Kategorie	mandatory
Zugriff	ro
PDO Mapping	nicht möglich
Wertebereich	2
Defaultwert	2

Subindex 1 wird für das Senden eines Zeichens an den Verstärker verwendet.

Subindex	**1**
Beschreibung	StdIn
Kategorie	mandatory
Zugriff	ro
PDO Mapping	möglich
Wertebereich	UNSIGNED8
Defaultwert	—

Subindex 2 wird für den Empfang eines Zeichens vom Verstärker verwendet.

Subindex	**2**
Beschreibung	StdOut
Kategorie	mandatory
Zugriff	w
PDO Mapping	möglich
Wertebereich	UNSIGNED8
Defaultwert	0

4.2.1.19 Objekt 2000h: Manufacturer Warnings

Dieses Objekt liefert Informationen über geräteinterne Warnungen.

Index	**2000_h**
Name	Manufacturer warnings
Objektcode	VAR
Datentyp	UNSIGNED32
Kategorie	optional

Zugriff	ro
PDO Mapping	möglich
Wertebereich	UNSIGNED32
Defaultwert	—

Bit-kodierte Warnungen:

Bit	Beschreibung
0	n 1: I²t-Meldeschwelle überschritten
1	n 2: Ballastleistung erreicht
2	n 3: Schleppfehler
3	n 4: Ansprechüberwachung aktiv
4	n 5: Netzphase fehlt
5	n 6: Software-Endschalter 1 hat angesprochen
6	n 7: Software-Endschalter 2 hat angesprochen
7	n 8: Fehlerhafter Fahrauftrag gestartet
8	n 9: Beim Fahrauftrag-Start war kein Referenzpunkt gesetzt
9	n 10: PSTOP aktiv
10	n 11: NSTOP aktiv
11	n 12: Motordefaultwerte wurden geladen
12	n 13: Erweiterungskarte arbeitet nicht ordnungsgemäß
13	n 14: Motorphase
14	n 15: fehlerhafter VCT-Eintrag angewählt
15	n 16: Summenwarnung für n17 bis n31
16	n 17: CAN-Sync ist nicht eingeloggt (mit SYNCSRC = 3)
17	n 18 -Multiturn overflow: Max. Anzahl von Umdrehungen überschritten
18 ... 30	n 19- 31: reserviert
31	n 32 -Firmware ist eine nicht freigegebene Beta-Version

* = Diese Warnmeldungen führen zu einem geführten Stillsetzen des Antriebs (Bremsung mit Notrampe)

4.2.1.20 Objekt 2014-2017h: Maske 1-4 für Sende-PDOs

Um die Buslast bei ereignisgesteuerten PDOs zu senken, kann die Überwachung für einzelne Bits im PDO mit Hilfe der Masken abgeschaltet werden. Beispielsweise kann so auch erreicht werden, dass Positionsistwerte immer nur einmal pro Umdrehung gemeldet werden.
Diese Objekte maskieren die PDO - Kanäle 1 bis 4. Sollten nur 2 Byte in einem PDO definiert sein, überlagert die Maske auch nur zwei Byte obwohl 4 Byte Maskeninformation übertragen worden ist. Ein aktiviertes Bit in der Maske bedeutet, dass die Überwachung für das korrespondierende Bit im PDO aktiv ist, d. h. überwacht wird.

Index	$0x2014_h$...$0x2017_h$.
Name	tx_mask 1-4
Objektcode	ARRAY
Anzahl Elemente	2
Datentyp	UNSIGNED32

Subindex	**1**
Kurzbeschreibung	tx_mask1-4_low
Mode	unabhängig
Zugriff	rw
PDO Mapping	nicht möglich
Einheit	—
Wertebereich	UNSIGNED32
EEPROM	nein
Defaultwert	$FFFFFFFF_h$

Subindex	**2**
Kurzbeschreibung	tx_mask1-4_high
Mode	unabhängig
Zugriff	rw
PDO Mapping	nicht möglich
Einheit	—
Wertebereich	UNSIGNED32
EEPROM	nein
Defaultwert	$FFFFFFFF_h$

4.2.1.21 Objekt 2030h: DP-RAM Variablen (nur schreibbar)

Index	**2030_h**
Name	DP-Ram Variablen, nur schreibbar (PDO)
Objektcode	ARRAY
Datentyp	INTEGER32
Kategorie	optional

Subindex	**0**
Name	Anzahl der Einträge
Datentyp	UNSIGNED8
Kategorie	optional
Zugriff	ro
PDO Mapping	nicht möglich
Wertebereich	9
Defaultwert	9

Subindex	**1...8**
Beschreibung	DP-Ram Variable 9...16
Kategorie	optional
Zugriff	rww
PDO mapping	möglich
Wertebereich	INTEGER32
Defaultwert	0

4.2.1.22 Objekt 2040h: Gearing factor for electronic gearing

Dieses Objekt definiert die Übersetzung für das elektronische Getriebe zwischen Master- und Slave-Verstärker, die über Inkrementalgeber (ROD) verbunden sind. Diese Objekte sind nur relevant für OPMODE 4 bzw. CANopen-Mode 0x84.

Index	**2040h**
Name	Electronic gearing factor
Objektcode	RECORD
Kategorie	optional

Subindex	**0**
Beschreibung	Anzahl der Einträge
Datentyp	UNSIGNED8
Kategorie	optional
Zugriff	ro
PDO Mapping	nicht möglich
Wertebereich	2
Defaultwert	2

Subindex 1 bezieht sich auf das Eingangssignal vom Master abhängig vom gewählten Feedback und GEARMODE.

Subindex	**1**
Beschreibung	Gearing Input
Kategorie	optional
Zugriff	rw
PDO Mapping	möglich
Wertebereich	INTEGER32
Defaultwert	1024

Subindex 2 liefert die Bewegung des Slave in Abhängigkeit von den Master Impulsen.

Subindex	**2**
Beschreibung	Gearing Output
Kategorie	optional
Zugriff	rw
PDO Mapping	möglich
Wertebereich	UNSIGNED32
Defaultwert	1

4.2.1.23 Objekt 2041h: Istwerte elektrisches Getriebe

Dieses Objekt dient zur Anzeige der eingelesenen Mastergeschwindigkeit, der internen Slave-Sollgeschwindigkeit als Richtgröße für den Slave des elektrischen Getriebes und dem internen Steuerwort für die Ein- / Aussynchronisierung des elektrischen Getriebes. Ein Beispiel zur Verwendung des elektrischen Getriebes finden Sie auf S.114.

Index	**2041$_h$**
Name	Istwerte elektrisches Getriebe
Objektcode	RECORD
Kategorie	optional

Subindex	**0**
Beschreibung	Anzahl der Einträge
Datentyp	UNSIGNED8
Kategorie	optional
Zugriff	ro
PDO Mapping	nicht möglich
Wertebereich	3
Defaultwert	3

Subindex 1 dient zur Anzeige des aus der Mastergeschwindigkeit über das Übersetzungsverhältnis (Objekt 2040h) ermittelten internen Soll-Slavegeschwindigkeit.

Subindex	**1**
Beschreibung	Slave-Geschwindigkeitssollwert
Kategorie	optional
Zugriff	ro
PDO Mapping	möglich
Wertebereich	INTEGER32
Defaultwert	0

Der Subindex 2 zeigt die ermittelte Mastergeschwindigkeit an.

Subindex	**2**
Beschreibung	Mastergeschwindigkeit des elektrischen Getriebes
Kategorie	optional
Zugriff	ro
PDO Mapping	möglich
Wertebereich	INTEGER32
Defaultwert	0

Subindex 3 dient zur Anzeige des Synchronisationszustandes der Slave-Achse des elektrischen Getriebs. Zur Bedeutung siehe Beispiel auf S.114.

Subindex	**3**
Beschreibung	Internes Kontrollwort für elektrisches Getriebe
Kategorie	optional
Zugriff	ro
PDO Mapping	möglich
Wertebereich	0, 1, 2, 5, 6
Defaultwert	0

4.2.1.24 Objekt 2051h: Konfiguration der Positionsregister

Dieses Objekt dient zur Freigabe, Einstellung der Triggerart und der Polarität der 16 Positionsregister. Über die Objekte 2052h und 2053h wird vorgegeben, ob die Position absolut oder relativ zur Istposition ausgewertet werden sollen.

Index	**2051h**
Name	Konfiguration der Positionsregister
Objektcode	ARRAY
Kategorie	optional

Subindex	**0**
Beschreibung	Anzahl der Einträge
Datentyp	UNSIGNED8
Kategorie	optional
Zugriff	ro
PDO Mapping	nicht möglich
Wertebereich	3
Defaultwert	3

Über den Subindex 1 können die 16 P1 .. P16 Positionsregister über die zugeordneten Bits 0..15 einzeln freigegeben oder gesperrt werden. Freigabe erfolgt mit Setzen des jewiligen Bits.

Subindex	**1**
Beschreibung	Freigabe der Positionsregister
Kategorie	optional
Zugriff	rww
PDO Mapping	möglich
Wertebereich	0...65535
Defaultwert	0

Subindex 2 definiert die Triggerart der 16 Positionsregister P1 - P16. Wird das entsprechende Bit auf 0 gesetzt, wird die Position ständig überwacht. Wird es auf 1 gesetzt, wird nach Ansprechen des Positionsregisters das entsprechende Freigabebit aus Subindex 1 zurückgesetzt und damit die Überwachung abgeschaltet.

Subindex	**2**
Beschreibung	Modus der Positionsregister
Kategorie	optional
Zugriff	rww
PDO Mapping	möglich
Wertebereich	0...65535
Defaultwert	0

Mit Hilfe des Subindex 3 wird die Polarität der Überwachung der 16 Positionsregister P1 - P16 eingestellt. Bei Einstellung auf 0 wird das Überschreiten der Position überwacht, bei 1 das Unterschreiten der Position des entsprechenden Positionsregisters.

Subindex	**3**
Beschreibung	Polarität der Positionsregister
Kategorie	optional
Zugriff	rw
PDO Mapping	möglich
Wertebereich	0...65535
Defaultwert	0

4.2.1.25 Objekt 2052h: Positionsregister, absolut

Über dieses Objekt werden die Werte der 16 Positionsregister P1 - P16 als absolute Positionsangaben eingetragen.

Index	**2052$_h$**
Name	Positionsregister, absolut
Objektcode	ARRAY
Kategorie	optional

Subindex	**0**
Beschreibung	Anzahl der Einträge
Datentyp	UNSIGNED8
Kategorie	optional
Zugriff	ro
PDO Mapping	nicht möglich
Wertebereich	16
Defaultwert	16

Über die Subindex 1 bis 16 werden die zu überwachenden Positionen der Positionsregister P1...P16 angesprochen.

Subindex	**1...16**
Beschreibung	Positionsregister 1..16, absolut
Kategorie	optional
Zugriff	rww
PDO Mapping	möglich
Wertebereich	INTEGER32
Defaultwert	0

4.2.1.26 Objekt 2053h: Positionsregister, relativ

Über dieses Objekt werden die Werte der 16 Positionsregister P1 - P16 als Positionsangaben relativ zur aktuellen Istposition eingetragen.

Index	**2053$_h$**
Name	Positionsregister, relativ
Objektcode	ARRAY
Kategorie	optional

Subindex	**0**
Beschreibung	Anzahl der Einträge
Datentyp	UNSIGNED8
Kategorie	optional
Zugriff	ro
PDO Mapping	nicht möglich
Wertebereich	16
Defaultwert	16

Über die Subindex 1 bis 16 werden die zu überwachenden Positionen der Positionsregister P1...P16 angesprochen.

Subindex	**1...16**
Beschreibung	Positionsregister 1..16, relativ
Kategorie	optional
Zugriff	rww
PDO Mapping	möglich
Wertebereich	INTEGER32
Defaultwert	0

4.2.1.27 Objekt 2061h: Strombegrenzung im Drehzahlmodus

Dieses Objekt dient zur schnellen Strombegrenzung im Drehzahlmode (0x3). Ein Wert von 3280 entspricht dabei dem maximalen Gerätestrom, der über DIPEAK abfragbar ist. Dieses Objekt wirkt auf den ASCII-Parameter DPRILIMIT. Damit es wirksam werden kann, muss der Konfigurationsparameter DILIM auf 1 stehen.

Index	**2061$_h$**
Name	Strombegrenzung im Drehzahlmodus
Objektcode	VAR
Datentyp	UNSIGNED16
Kategorie	optional

Zugriff	rww
PDO Mapping	möglich
Wertebereich	0 .. 3280
Defaultwert	0

4.2.1.28 Objekt 2080h: Motion task for profile position mode

Dieses Objekt ist eine Erweiterung des Profile Position Mode. Wenn der Wert ungleich 0 ist, wird der adressierte Fahrsatz mit der nächsten steigenden Flanke des Bit "New Setpoint" (Bit 4) des Controlword gestartet, sofern auch das Bit "Change Set Immediately" (Bit 5) gesetzt ist.

Nach Start des Fahrsatzes wird der Wert des Objekts automatisch wieder auf 0 zurückgesetzt.

Index	**2080$_h$**
Name	Motion tasks in PP-Mode
Objektcode	VAR
Datentyp	UNSIGNED16
Kategorie	optional

Zugriff	rw
PDO Mapping	möglich
Wertebereich	1 ... 300
Defaultwert	0

4.2.1.29 Objekt 2081h: Active motion task display

Dieses Objekt zeigt den letzten Fahrauftrag an, der im Verstärker gestartet wurde. Fahrsatznummern von 1 bis 200 zeigen Flash-EEProm Fahraufträge, Nummern von 201 bis 300 zeigen RAM Fahraufträge an.

Wenn Objekt 2080$_h$ keinen Wert beinhaltet und ein Fahrauftrag über den "New-setpoint/setpoint acknowledge" Mechanismus des Profile Position Mode gestartet wurde, wird Fahrauftrag 0 verwendet und angezeigt.

Wenn Sie eine Serie von Folgefahraufträgen starten (Bit 3 des Fahrauftrag Control Word O_C, Objekt 35B9 sub1 gesetzt), wird der aktive Fahrauftrag angezeigt.

Index	**2081$_h$**
Name	Active motion task display
Objektcode	VAR
Datentyp	UNSIGNED16
Kategorie	optional

Zugriff	rw
PDO Mapping	möglich
Wertebereich	1 ... 300
Defaultwert	0

4.2.1.30 Objekt 2082h: Copy motion tasks

Mit diesem Objekt können Fahraufträge im Servoverstärker kopiert werden. Der Fahrauftrag, der im Low Word addressiert ist, wird zu dem Fahrsatz kopiert, der im High Word addressiert ist.

INFO **EEPROM Fahraufträge 1...200 können nur bei gesperrter Endstufe geschrieben werden.**

Index	**2082$_h$**
Name	Copy motion tasks
Objektcode	VAR
Datentyp	UNSIGNED32
Kategorie	optional

Zugriff	wo
PDO Mapping	nicht möglich
Wertebereich	High Word: 0 .. 300, low word: 0 .. 300
Defaultwert	-

4.2.1.31 Objekt 2083h: Delete Motion tasks

Das Objekt ermöglicht das Löschen aller Flash-EEPROM Fahrsätze. Dieser Vorgang kann nur gestartet werden, wenn eine spezielle Signatur ("prom") geschrieben wurde.
"prom" muss als Unsigned32 - Zahl 6D6F7270$_h$ übertragen werden.
Der Löschvorgang ist nur möglich, wenn die Endstufe gesperrt ist und der NMT-Status PREOPERATIONAL ist.

Index	**2083$_h$**
Name	Delete motion tasks
Objektcode	VAR
Datentyp	UNSIGNED32
Kategorie	optional

Zugriff	wo
PDO Mapping	nicht möglich
Wertebereich	UNSIGNED32
Defaultwert	-

4.2.1.32 Objekt 2090h: DP-RAM Variablen (nur lesbar)

Index	**2090$_h$**
Name	DP-Ram Variablen, nur lesbar (PDO)
Objektcode	ARRAY
Datentyp	INTEGER32
Kategorie	optional

Subindex	**0**
Name	Anzahl der Einträge
Datentyp	UNSIGNED8
Kategorie	optional
Zugriff	ro
PDO Mapping	nicht möglich
Wertebereich	9
Defaultwert	9

Subindex	**1...8**
Beschreibung	DP-Ram Variable 9...16
Kategorie	optional
Zugriff	ro
PDO mapping	möglich
Wertebereich	INTEGER32
Defaultwert	0

4.2.1.33 Objekt 20A0h: Latchposition 1, positive Flanke

Über dieses Objekt wird die Position ausgelesen, an der die erste positive Flanke am digitalen Eingang 1 nach der Latchfreigabe (s.a. Objekt20A4) aufgetreten war. Durch das Auslesen wird die Latchmimik wieder für einen Vorgang aktiv geschaltet. Voraussetzung ist die Konfiguration des Eingangs 1 als Latcheingang (IN1MODE 26).

Index	**20A0h**
Name	Latchposition 1, positive Flanke
Objektcode	VAR
Datentyp	INTEGER32
Kategorie	optional

Zugriff	ro
PDO Mapping	möglich
Wertebereich	INTEGER32
Defaultwert	0

4.2.1.34 Objekt 20A1h: Latchposition 1, negative Flanke

Über dieses Objekt wird die Position ausgelesen, an der die erste negative Flanke am digitalen Eingang 1 nach der Latchfreigabe (s.a. Objekt 20A4) aufgetreten war. Durch das Auslesen wird die Latchmimik wieder für einen Vorgang aktiv geschaltet. Voraussetzung ist die Konfiguration des Eingangs 1 als Latcheingang (IN1MODE 26).

Index	**20A1h**
Name	Latchposition 1, negative Flanke
Objektcode	VAR
Datentyp	INTEGER32
Kategorie	optional

Zugriff	ro
PDO Mapping	möglich
Wertebereich	INTEGER32
Defaultwert	0

4.2.1.35 Objekt 20A2h: Latchposition 2, positive Flanke

Über dieses Objekt wird die Position ausgelesen, an der die erste positive Flanke am digitalen Eingang 2 nach der Latchfreigabe (s.a. Objekt20A4) aufgetreten war. Durch das Auslesen wird die Latchmimik wieder für einen Vorgang aktiv geschaltet. Voraussetzung ist die Konfiguration des Eingangs 2 als Latcheingang (IN2MODE 26).

Index	**20A2h**
Name	Latchposition 2, positive Flanke
Objektcode	VAR
Datentyp	INTEGER32
Kategorie	optional

Zugriff	ro
PDO Mapping	möglich
Wertebereich	INTEGER32
Defaultwert	0

4.2.1.36 Objekt 20A3h: Latchposition 2, negative Flanke

Über dieses Objekt wird die Position ausgelesen, an der die erste negative Flanke am digitalen Eingang 2 nach der Latchfreigabe (s.a. Objekt 20A4) aufgetreten war. Durch das Auslesen wird die Latchmimik wieder für einen Vorgang aktiv geschaltet. Voraussetzung ist die Konfiguration des Eingangs 2 als Latcheingang (IN2MODE 26).

Index	**20A3$_h$**
Name	Latchposition 2, negative Flanke
Objektcode	VAR
Datentyp	INTEGER32
Kategorie	optional

Zugriff	ro
PDO Mapping	möglich
Wertebereich	INTEGER32
Defaultwert	0

4.2.1.37 Objekt 20A4h: Latch-Kontrollregister

Das Latch-Kontrollregister dient zum Freischalten der Latch-Überwachung der Digitaleingänge 1 und 2. Freigeschaltet wird mit einem 1-Signal, gesperrt mit einem 0-Signal. Ob ein Latchereignis aufgetreten ist, lässt sich über den Herstellerstatus (siehe Tabelle), Objekt 1002, erkennen.

Index	**20A4$_h$**
Name	Latch-Kontrollregister
Objektcode	VAR
Datentyp	UNSIGNED8
Kategorie	optional

Zugriff	rww
PDO Mapping	möglich
Wertebereich	0...15
Defaultwert	0

Bit	Flanke	Bit im Objekt 1002
0	Freigabe der positiven Flanke an Digitaleingang 1	20
1	Freigabe der negativen Flanke an Digitaleingang 1	21
2	Freigabe der positiven Flanke an Digitaleingang 2	4
3	Freigabe der negativen Flanke an Digitaleingang 2	18
4-7	reserviert	

4.2.1.38 Objekt 20B0h: Triggervariable Digitaleingang 20

Über dieses Objekt kann die Triggervariable für den Softwareeingang 20 eingestellt werden. Dieses auch mappbare Objekt kann für den Start-/Stop-Vorgang im elektrischen Getriebe verwendet werden (bei IN20MODE = 51 oder 53).

Index	**$20B0_h$**
Name	Triggervariable Digitaleingang 20
Objektcode	VAR
Datentyp	INTEGER32
Kategorie	optional

Zugriff	rww
PDO Mapping	möglich
Wertebereich	INTEGER32
Defaultwert	0

4.2.1.39 Objekt 20B1h: Steuerwort Digitaleingänge 5...20

Über die unteren 16 Bits dieses Objekt können die digitalen Softwareeingänge 5 bis 20 angesteuert werden.

Index	**$20B1_h$**
Name	Steuerwort Digitaleingänge 5 .. 20
Objektcode	VAR
Datentyp	UNSIGNED32
Kategorie	optional

Zugriff	ro
PDO Mapping	möglich
Wertebereich	0...65535
Defaultwert	0

4.2.1.40 Objekt 20B2h: Analoge Eingänge

Mit Hilfe dieses Objekts können die Spannungen der Analogeingänge 1 und 2 eingelesen werden.

Index	**$20B2_h$**
Name	Analoge Eingänge
Objektcode	ARRAY
Datentyp	INTEGER16
Kategorie	optional

Subindex	**0**
Name	Anzahl der Einträge
Datentyp	UNSIGNED8
Kategorie	optional
Zugriff	ro
PDO Mapping	nicht möglich
Wertebereich	2
Defaultwert	2

Subindex 1 liefert den Digitalwert des Analogeingangs 1. (10 V ~ 2000h).

Subindex	**1**
Beschreibung	Spannung Analogeingang 1
Kategorie	optional
Zugriff	ro
PDO mapping	möglich
Wertebereich	INTEGER16
Defaultwert	-

Subindex 2 liefert den Digitalwert des Analogeingangs 2. (10 V ~ 2000h).

Subindex	**2**
Beschreibung	Spannung Analogeingang 2
Kategorie	optional
Zugriff	ro
PDO mapping	möglich
Wertebereich	INTEGER16
Defaultwert	-

4.2.1.41 Objekt 2100h: Schreib-Dummy

Index	**2100$_h$**
Name	Schreib-Dummy-Variablen für Mappingzwecke
Objektcode	RECORD
Datentyp	UNSIGNED32
Kategorie	optional

Zugriff	rww
PDO Mapping	möglich
Wertebereich	UNSIGNED32
Defaultwert	1

4.2.1.42 Objekt 2101h: Lese-Dummy

Index	**2101$_h$**
Name	Lese-Dummy-Variablen für Mappingzwecke
Objektcode	RECORD
Datentyp	UNSIGNED32
Kategorie	optional

Zugriff	ro
PDO Mapping	möglich
Wertebereich	UNSIGNED32
Defaultwert	1

4.2.1.43 Objekt 60FDh: Digital inputs (DS402)

Dieser Index definiert einfache digitale Eingänge für Servoverstärker. Die Bits 0 bis 2 können vom Servoverstärker bedient werden, wenn die benötigte Funktion dem digitalen Eingang mit dem ASCII-Kommando INxMODE (x von 1 bis 4) zugeordnet wurde, z.B. IN3MODE = 2, PSTOP - Function, siehe ASCII-Dokumentation.

Index	**60FD$_h$**
Name	digital inputs
Objektcode	VAR
Datentyp	UNSIGNED32
Kategorie	optional

Zugriff	ro
PDO Mapping	möglich
Wertebereich	UNSIGNED32
Defaultwert	0

Daten Beschreibung:

31 16	15 4	3	2	1	0
hersteller-spezifisch	**interlock**	**interlock**	**Referenz-schalter**	**pos. Endschalter**	**neg. Endschalter**
MSB					LSB

Die Schalter müssen als Schließer ausgelegt sein (active high).

4.2.1.44 Objekt 6502h: Supported drive modes (DS402)

Ein Servoverstärker unterstützen mehr als einen und verschiedene unterschiedliche Betriebsarten. Dieses Objekt gibt einen Überblick über die möglichen Betriebsarten des Servoverstärkers.

Index	**6502_h**
Name	supported drive modes
Objektcode	VAR
Datentyp	UNSIGNED32
Kategorie	optional

Zugriff	ro
PDO Mapping	möglich
Wertebereich	UNSIGNED32
Defaultwert	0x6D (ip hm tq pv pp)

Daten Beschreibung:

31 ... 16	15 ... 7	6	5	4	3	2	1	0
herstellerspezifisch	**reserviert**	**ip**	**hm**	**reserviert**	**tq**	**pv**	**vl**	**pp**
MSB								LSB

4.3 PDO Konfiguration

PDOs werden für die Prozess Daten Kommunikation verwendet.

Es gibt zwei unterschiedliche Typen von PDOs: Empfangs PDOs (RPDOs) und Sende PDOs (TPDOs). Der Inhalt der PDOs ist vordefiniert (siehe Beschreibung auf den Seiten 50 und 52). Wenn der Dateninhalt für eine spezielle Anwendung nicht brauchbar ist, können die Datenobjekte in den PDOs frei geändert (remapped) werden.

Dateneintrag in den PDOs

MSB		LSB
Index (16 Bit)	Subindex (8 Bit)	Datenlänge in Bits (8 Bit)

Konfigurationsprozedur für ein freies Mapping (Beispiel für TPDO1)

1. Aktuelles Mapping löschen: Schreiben von 0 in Subindex 0 des Objekts

COB-ID	Control byte	Index		Sub-index	Daten	Kommentar
		Low byte	High byte			
601	2F	00	1A	00_h	00 00 00 00	Lösche aktuelles Mapping

2. Mapping aufbauen mit Objekten des Objektverzeichnisses (siehe S.97), die gemappt werden dürfen, z.B.

COB-ID	Control byte	Index		Sub-index	Daten	Kommentar
		Low byte	High byte			
601	23	00	1A	01_h	10 00 41 60	Erster Eintrag: CANopen Statuswort mit 16 Bits
601	23	00	1A	02h	20 00 02 10	Zweiter Eintrag: Herstellerstatus mit 32 Bits

3. Schreibe die Anzahl der gemappten Objekte in Subindex 0 des Objekts.

COB-ID	Control byte	Index		Sub-index	Daten	Kommentar
		Low byte	High byte			
601	2F	00	1A	00_h	02 00 00 00	Prüfe korrekte Anzahl Einträge

Mapping sollte immer beendet sein, bevor das Network Management auf OPERATIONAL geschaltet wird.

4.3.1 Receive PDOs (RXPDO)

Vier Empfangs PDOs können im Servoverstärker konfiguriert werden:
- Konfigurieren der Kommunikation (1400-1403$_h$)
- Konfigurieren des PDO Inhalts (Mapping, Objekte 1600-1603$_h$)

4.3.1.1 Objekt 1400-1403h: 1. - 4. RXPDO Kommunikationsparameter (DS301)

Index	**1400$_h$ ... 1403$_h$ for RXPDO 1 ... 4**
Name	receive PDO parameter
Objekt Code	RECORD
Datentyp	PDO CommPar
Kategorie	mandatory

Definierte Sub-Indizes

Subindex	**0**
Name	Anzahl der Einträge
Datentyp	UNSIGNED8
Kategorie	mandatory
Zugriff	ro
PDO Mapping	nicht möglich
Wertebereich	2
Defaultwert	2

Subindex	**1**
Name	vom PDO verwendete COB-ID
Kategorie	mandatory
Zugriff	rw
PDO Mapping	nicht möglich
Wertebereich	UNSIGNED32
Defaultwert	Index 1400$_h$: 200$_h$ + Node-ID Index 1401$_h$: 300$_h$ + Node-ID Index 1402$_h$: 400$_h$ + Node-ID Index 1403$_h$: 500$_h$ + Node-ID

Subindex	**2**
Name	Übertragungstyp
Kategorie	mandatory
Zugriff	rw
PDO Mapping	nicht möglich
Wertebereich	UNSIGNED8
Defaultwert	FF$_h$

Subindex 1 beinhaltet die COB-Id des PDO als Bit kodierte Information:

Bit	Wert	Bedeutung
31	0	PDO existiert / ist gültig
	1	PDO existiert nicht / ist nicht gültig
30	0	RTR erlaubt auf diesem PDO, nicht unterstützt
	1	RTR nicht erlaubt auf diesem PDO, nicht unterstützt
29	0	11 Bit-ID (CAN 2.0A)
	1	29 Bit-ID (CAN 2.0B), nicht unterstützt
28 .. 11	X	Identifier-Bits mit 29 Bit-ID, nicht relevant
	0	wenn Bit 29=0
10 .. 0	X	Bits 10-0 der COB-ID

Subindex 2 beinhaltet den Übertragungstyp des PDO. Es gibt zwei Einstellarten:

— Wert FF$_h$ oder 255 für Ereignis gesteuertes PDO, das direkt vom Empfänger interpretiert und ausgeführt wird.

— Werte von 0 bis 240 führen zu einer mit dem SYNC Telegramm kontrollierten Interpretation des PDO Inhalts. Werte von 1 bis 240 bedeuten, dass 0 bis 239 SYNC-Telegramme ignoriert werden, bevor eines interpretiert wird. Der Wert 0 bedeutet, dass das nächste SYNC-Telegramm interpretiert wird.

4.3.1.2 Objekt 1600-1603h: 1. - 4. RXPDO Mapping Parameter (DS301)

Index	**1600_h - 1603_h for RXPDO 1 .. 4**
Name	receive PDO mapping
Objekt Code	RECORD
Datentyp	PDO Mapping
Kategorie	mandatory

Subindex	**0**
Name	number of mapped application objects in PDO
Datentyp	UNSIGNED8
Kategorie	mandatory
Zugriff	rw
PDO Mapping	nicht möglich
Wertebereich	0: PDO ist nicht aktiv 1 - 8: PDO aktiviert, Mappings werden nur Byte-weise übernommen
Defaultwert	PDO1: 1 PDO2: 2 PDO3: 2 PDO4: 2

Subindex	**1 - 8**
Name	PDO - mapping for the n-th application object
Kategorie	conditional; hängt von Anzahl und Größe der gemappten Objekte ab.
Zugriff	rw
PDO Mapping	nicht möglich
Wertebereich	UNSIGNED32
Defaultwert	Siehe unten

4.3.1.3 Default RXPDO Definition

RXPDO 1:

Subindex	Wert	Bedeutung
0	1	Ein PDO Mapping Eintrag
1	60 40 00 10	Controlword

RXPDO 2:

Subindex	Wert	Bedeutung
0	2	Zwei PDO Mapping Einträge
1	60 40 00 10	Controlword
2	60 60 00 08	Betriebsart

RXPDO 3:

Subindex	Wert	Bedeutung
0	2	Zwei PDO Mapping Einträge
1	60 40 00 10	Controlword
2	60 7A 00 20	Zielposition (Betriebsart PP)

RXPDO 4:

Subindex	Wert	Bedeutung
0	2	Zwei PDO Mapping Einträge
1	60 40 00 10	Controlword
2	60 FF 00 20	Zielgeschwindigkeit (Betriebsart PV)

4.3.2 Transmit PDOs (TXPDO)

Vier Sende PDOs können im Servoverstärker konfiguriert werden:
- Konfigurieren der Kommunikation (Objekte 1800-1803$_h$)
- Konfigurieren des PDO Inhalts (Mapping, Objekte 1A00-1A03$_h$)

4.3.2.1 Objekt 1800-1803h: 1. - 4. TXPDO Kommunikation Parameter (DS301)

Index	**1800$_h$ ... 1803$_h$ for TXPDO 1 ... 4**
Name	transmit PDO parameter
Objekt Code	RECORD
Datentyp	PDO CommPar
Kategorie	mandatory

Subindex	**0**
Name	Anzahl der Einträge
Datentyp	UNSIGNED8
Kategorie	mandatory
Zugriff	ro
PDO Mapping	nicht möglich
Wertebereich	5
Defaultwert	5

Subindex	**1**	
Name	COB-ID used by PDO	
Kategorie	mandatory	
Zugriff	rw	
PDO Mapping	nicht möglich	
Wertebereich	UNSIGNED32	
Defaultwert	Index 1800$_h$: 180$_h$ + Node-ID Index 1802$_h$: 380$_h$ + Node-ID	Index 1801$_h$: 280$_h$ + Node-ID Index 1803$_h$: 480$_h$ + Node-ID

Subindex	**2**
Name	transmission type
Kategorie	mandatory
Zugriff	rw
PDO Mapping	nicht möglich
Wertebereich	UNSIGNED8
Defaultwert	FF$_h$

Subindex	**3**
Name	inhibit time
Kategorie	optional
Zugriff	rw
PDO Mapping	nicht möglich
Wertebereich	UNSIGNED16 (n*1/10ms)
Defaultwert	0$_h$

Subindex	**4**
Name	reserviert
Kategorie	optional
Zugriff	rw
PDO Mapping	nicht möglich
Wertebereich	0
Defaultwert	0

Subindex	**5**
Name	event timer
Kategorie	optional
Zugriff	rw
PDO Mapping	nicht möglich
Wertebereich	UNSIGNED16 (0=nicht verwendet, n*1/10ms)
Defaultwert	0$_h$

Subindex 1 beinhaltet die COB-Id des PDO als Bit kodierte Information:

Bit	Wert	Bedeutung
31	0	PDO existiert / ist gültig
	1	PDO existiert nicht / ist nicht gültig
30	0	RTR erlaubt auf diesem PDO, nicht unterstützt
	1	RTR nicht erlaubt auf diesem PDO, nicht unterstützt
29	0	11 Bit-ID (CAN 2.0A)
	1	29 Bit-ID (CAN 2.0B), nicht unterstützt
28 .. 11	X	Identifier-Bits mit 29 Bit-ID, nicht relevant
	0	Wenn Bit 29=0
10 .. 0	X	Bits 10-0 der COB-ID

Subindex 2 beinhaltet den Übertragungstyp des PDO. Es gibt zwei Einstellarten:

— Wert **FF$_h$ oder 255** für ein ereignisgesteuertes PDO, das sofort nach einer Änderung in den gemappten Objekten gesendet wird. Die EinstellungenSetting von Subindex 3 und 5 haben einen Einfluss auf die Sendung des PDO. Mit **Subindex 3** können Sie konfigurieren, nach welcher Mindestzeit das PDO gesendet wird, wenn sich der PDO-Dateninhalt geändert hat (Reduktion der Busbelastung). Mit **Sub_index 5** wird ein Timer verwendet, der nach jedem Ereignis gesteuerten Senden des PDO neu gestartet wird. Auch wenn der PDO Inhalt sich nicht geändert hat, wird das PDO angestoßen vom Event Timer gesendet.

— Werte von **0 bis 240** führen zu vom SYNC Telegramm kontrolliertem Senden des PDO. Werte von 1 bis 240 definieren,wie oft das SYNC-Telegram zur Sendung des PDO führt. 0 bedeutet, dass das nächste SYNC-Telegramm zum Senden des PDO führt.

4.3.2.2 Objekt 1A00-1A03h: 1. - 4. TXPDO Mapping Parameter (DS301)

Index	**1A00$_h$ - 1A03$_h$ for TXPDO 1 .. 4**
Name	transmit PDO mapping
Objekt Code	RECORD
Datentyp	PDO Mapping
Kategorie	mandatory

Subindex	**0**
Name	number of mapped application objects in PDO
Datentyp	UNSIGNED8
Kategorie	mandatory
Zugriff	rw
PDO Mapping	nicht möglich
Wertebereich	0: PDO ist nicht aktiv 1 - 8: PDO aktiviert, Mappings werden nur Byte-weise übernommen
Defaultwert	PDO1: 1 PDO2: 2 PDO3: 2 PDO4: 2

Subindex	**1 - 8**
Name	PDO - mapping for the n-th application object
Kategorie	conditional; hängt von Anzahl und Große der gemappten Objekte ab.
Zugriff	rw
PDO Mapping	nicht möglich
Wertebereich	UNSIGNED32
Defaultwert	Siehe unten

4.3.2.3 Default TXPDO Definition

TXPDO 1:

Subindex	Wert	Bedeutung
0	1	Ein PDO-Mapping Einträag
1	60 41 00 10	Statusword

TXPDO 2:

Subindex	Wert	Bedeutung
0	2	Zwei PDO-Mapping Einträge
1	60 41 00 10	Statusword
2	60 61 00 08	Anzeige der Betriebsart

TXPDO 3:

Subindex	Wert	Bedeutung
0	2	Zwei PDO-Mapping Einträge
1	60 41 00 10	Statusword
2	60 64 00 20	Positions-Istwert

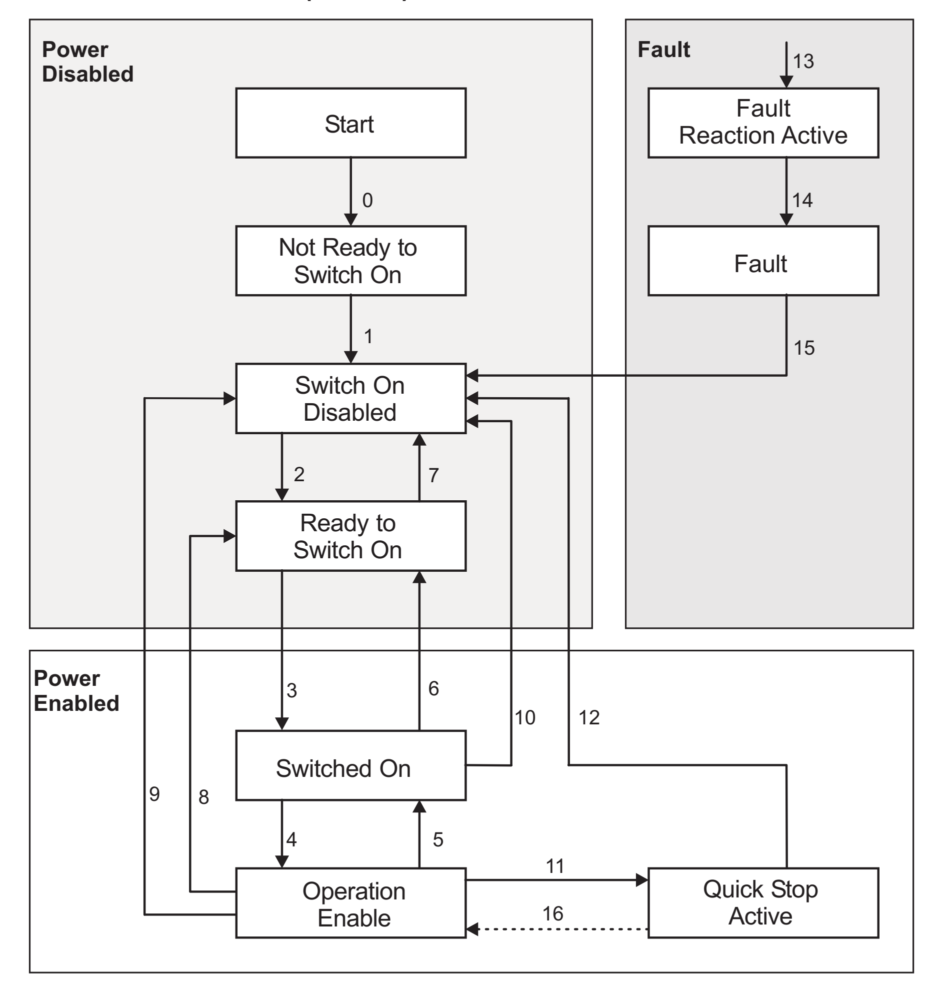
TXPDO 4:

Subindex	Wert	Bedeutung
0	2	Zwei PDO-Mapping Einträge
1	60 41 00 10	Statusword
2	60 6C 00 20	Geschwindigkeits-Istwer

4.4 Gerätesteuerung (dc)

Mit Hilfe der S300/S700 Gerätesteuerung können sämtliche Fahrfunktionen in den entsprechenden Betriebsarten ausgeführt werden. Die Steuerung des S300/S700 ist über eine modeabhängige Zustandsmaschine realisiert. Zur Steuerung der Zustandsmaschine dient das controlword (⇒ S. 57). Die Betriebsarteinstellung erfolgt über das Objekt "Modes of Operation" (⇒ S. 97). Die Zustände der Zustandsmaschine können mit dem statusword ermittelt werden (⇒ S. 59).

4.4.1 Zustandsmaschine (DS402)



4.4.1.1 Zustände der Zustandsmaschine

Zustand	Beschreibung
Not Ready to Switch On	S300/S700 ist nicht einschaltbereit, es wird keine Betriebsbereitschaft (BTB) vom Reglerprogramm gemeldet.
Switch On Disable	S300/S700 ist einschaltbereit, Parameter können übertragen werden, Zwischenkreis-spannung kann eingeschaltet werden, Fahrfunktionen können noch nicht ausgeführt werden
Ready to Switch On	Zwischenkreisspannung kann angeschaltet werden. Parameter können übertragen werden, Fahrfunktionen können noch nicht ausgeführt werden
Switched On	Zwischenkreisspannung muss angeschaltet sein, Parameter können übertragen werden, Fahrfunktionen können noch nicht ausgeführt werden, Endstufe ist freigegeben (Enabled)
Operation Enable	kein Fehler steht an, Endstufe ist freigegeben (Enabled), Fahrfunktionen sind freigegeben
Quick Stop Active	Antrieb ist mit der Notbremsrampe gestoppt worden, Endstufe ist freigegeben, Fahrfunktionen sind nicht freigegeben.
Fault Reaction Active	Ein Fehler ist aufgetreten und der Antrieb wird mit der Rampe Quickstop gestoppt.
Fault	Ein Fehler liegt an, der Antrieb wurde gestoppt und gesperrt.

4.4.1.2 Übergänge der Zustandsmaschine

Die Zustandsübergänge werden durch interne Ereignisse (z.B. Ausschalten der Zwischenkreisspannung) und durch die Flags im controlword (Bits 0,1,2,3,7) beeinflusst.

Übergang	Ereignis	Aktion
0	Reset	Initialisierung
1	Initialisierung erfolgreich abgeschlossen. S300/S700 betriebsbereit	keine
2	Bit 1 (disable Voltage) und Bit 2 (Quick Stop) im controlword gesetzt (‘Shutdown’ command). Zwischenkreisspannung kann anliegen.	keine
3	Bit 0 wird zusätzlich gesetzt (‘Switch On’ command)	Endstufe wird freigegeben (enable), sofern Hardware-enable anliegt (UND-Verknüpfung). Antrieb hat Drehmoment.
4	Bit 3 wird zusätzlich gesetzt (‘Enable Operation’ command)	Fahrfunktion in Abhängigkeit des eingestellten Mode wird freigegeben
5	Bit 3 wird gelöscht (‘Disable Operation’ command)	Fahrfunktion wird gesperrt. Antrieb wird mit der relevanten Rampe (modeabhängig) gestoppt. Aktuelle Position wird gehalten
6	Bit 0 wird gelöscht (‘Shutdown’ command)	Endstufe wird gesperrt (disable). Motor wird drehmomentfrei
7	Bit 1/2 werden gelöscht (’Quickstop’ / ‘Disable Voltage’ command)	keine
8	Bit 0 wird gelöscht (‘Shutdown’ command)	Endstufe wird gesperrt (disable) Motor wird drehmomentfrei
9	Bit 1 wird gelöscht (‘Disable Voltage’ command)	Endstufe wird gesperrt (disable) Motor wird drehmomentfrei
10	Bit 1/2 werden gelöscht (’Quickstop’ / ‘Disable Voltage’ command)	Endstufe wird gesperrt (disable). Motor wird drehmomentfrei.
11	Bit 2 wird gelöscht (’Quickstop’ command)	Antrieb wird mit der Notbremsrampe angehalten. Die Endstufe bleibt "enabled". Sollwerte werden gelöscht (Fahrsatznummer, digitaler Sollwert, Geschwindigkeit für Tippbetrieb oder Referenzfahrt). Vor dem erneuten Ausführen einer Fahrfunktion muss Bit 2 wieder gesetzt werden
12	Bit 1 wird gelöscht (‘Disable Voltage’ command)	Endstufe wird abgeschaltet (disable) Motor wird drehmomentfrei
13	Fehler Reaktion aktiv	Fehlerreaktion wird ausgeführt
14	Fehler Reaktion beendet	Antriebsfunktion ist deaktiviert. Endstufe kann ausgeschaltet werden.
15	"Fault Reset" Kommando vom Host empfangen	Ein Fehler-Reset wird ausgeführt, wenn aktuell kein Fehler im Servoverstäker avorliegt. Wenn der Status "Fault" verlassen wird, muss das Bit7 "Reset Fault" des Kontrollwortes vom Host gelöscht werden.
16	Bit 2 wird gesetzt	Fahrfunktion ist wieder freigegeben.

INFO

Wenn der Servoverstärker über das controlword / statusword bedient wird, dürfen keine Steuerkommandos über einen anderen Kommunikationskanal (RS232, CANopen, ASCII-Kanal, Optionskarte) gesendet werden.

4.4.2 Objektbeschreibung

4.4.2.1 Objekt 6040h: Controlword (DS402)

Aus der logischen Verknüpfung der Bits im controlword und externen Signalen (Freigabe der Endstufe) ergeben sich die Steuerkommandos. Die Definition der Bits wird im Folgenden beschrieben:

Index	**6040h**
Name	control word
Objektcode	VAR
Datentyp	UNSIGNED16

Zugriff	rw
PDO Mapping	möglich
Einheit	—
Wertebereich	0..65535
EEPROM	nein
Defaultwert	0

Bitbelegung im control word

Bit	Name	Bit	Name
0	Switch on	8	Halt (Zwischenstopp)
1	Disable Voltage	9	reserviert
2	Quick Stop	10	reserviert
3	Enable Operation	11	reserviert
4	Betriebsartspezifisch	12	reserviert
5	Betriebsartspezifisch	13	Herstellerspezifisch
6	Betriebsartspezifisch	14	Herstellerspezifisch
7	Reset Fault)*	15	Herstellerspezifisch

)* Über Bit 7 (Reset Fault) werden auch die Warnungen Schleppfehler (n03) und Ansprechüberwachung (Nodeguarding / Heartbeat-Fehler) gelöscht.

Kommandos im control word

Command	Bit 7 Fault Reset	Bit 3 Enable Operation	Bit 2 Quick Stop	Bit 1 Disable Voltage	Bit 0 Switch on	Übergänge
Shutdown	X	X	1	1	0	2, 6, 8
Switch on	X	X	1	1	1	3
Disable Voltage	X	X	X	0	X	7, 9, 10, 12
Quick Stop	X	X	0	1	X	7, 10, 11
Disable Operation	X	0	1	1	1	5
Enable Operation	X	1	1	1	1	4, 16
Fault Reset	1	X	X	X	X	15

Mit X gekennzeichnete Bits sind irrelevant.

Modeabhängige Bits im controlword

Die folgende Tabelle beschreibt die modeabhängigen Bits im controlword. Es werden z. Zt. nur herstellerspezifische Modi unterstützt. Die einzelnen Modi werden mit dem Objekt 6060_h "Modes of operation” eingestellt.

Operation mode	Nr.	Bit 4	Bit 5	Bit 6
Lage	**88h**	reserviert	reserviert	reserviert
Drehzahl digital	**80h**	reserviert	reserviert	reserviert
Strom digital	**82h**	reserviert	reserviert	reserviert
Drehzahl analog	**81h**	reserviert	reserviert	reserviert
Strom analog	**83h**	reserviert	reserviert	reserviert
Profile Position Mode (pp)	**01_h**	new_set_point	change_set_immediately	absolut / relativ
Profile Velocity Mode (pv)	**03_h**	reserviert	reserviert	reserviert
Profile Torque Mode (tq)	**04h**	reserviert	reserviert	reserviert
Homing Mode (hm)	**06_h**	homing_operation_start	reserviert	reserviert
Interpolated Position Mode (ip)	**07h**		reserviert	reserviert

Beschreibung der restlichen Bits im controlword

Im Folgenden werden die restlichen Bits des controlword beschrieben.

Bit 8 Zwischenstopp Ist Bit 8 gesetzt, wird in sämtlichen Modi der Antrieb gestoppt (Zwischenstopp). Die Sollwerte (Geschwindigkeit für Referenzfahrt oder Tippbetrieb, Fahrauftragsnummer, Sollwerte für Digitalmode) der einzelnen Modi bleiben erhalten
Bit 9,10 Diese Bits sind durch das Antriebsprofil (DS402) reserviert.
Bit 13, 14, 15 Diese Bits sind herstellerspezifisch und z.Zt. reserviert.

4.4.2.2 Objekt 6041h: Statusword (DS402)

Der momentane Zustand der Zustandsmaschine kann mit Hilfe des statusword (⇒ S. 36) abgefragt werden.

Index	**6041$_h$**
Name	Statusword
Objektcode	VAR
Datentyp	UNSIGNED16

Zugriff	rw
PDO Mapping	möglich
Einheit	—
Wertebereich	0..65535
EEPROM	ja
Defaultwert	0

Bitbelegung im Statuswort

Bit	Name	Bit	Name
0	Ready to switch on	8	Herstellerspezifisch (reserviert)
1	Switched on	9	Remote (immer 1)
2	Operation enable	10	Target reached
3	Fault	11	Internal limit active
4	Voltage enabled	12	Betriebsartspezifisch (reserviert)
5	Quick stop	13	Betriebsartspezifisch (reserviert)
6	Switch on disabled	14	Herstellerspezifisch (reserviert)
7	Warning	15	Herstellerspezifisch (reserviert)

Zustände der Statusmaschine

State	Bit 6 switch on disable	Bit 5 quick stop	Bit 3 fault	Bit 2 operation enable	Bit 1 switched on	Bit 0 ready to switch on
Not ready to switch on	1	X	0	0	0	0
Switch on disabled	1	X	0	0	0	0
Ready to switch on	0	1	0	0	0	1
Switched on	0	1	0	0	1	1
Operation enabled	0	1	0	1	1	1
Fault	0	X	1	0	0	0
Fault reaction active	0	X	1	1	1	1
Quick stop active	0	0	0	1	1	1

Die mit X gekennzeichneten Bits sind irrelevant

Beschreibung der restlichen Bits im Statuswort

Bit 4: voltage_enabled Die Zwischenkreisspannung liegt an, wenn das Bit gesetzt ist.

Bit 7: warning Für das Setzen von Bit 7 kann es mehrere Gründe geben, die zu dieser Warnung geführt haben. Der Grund für diese Warnung kann anhand des Objekts 2000$_h$ "herstellerspezifische Warnungen" ermittelt werden.

Bit 9: remote ist immer auf 1 gesetzt, d.h. der Antrieb kann immer kommunizieren und über das RS232-Interface beeinflusst werden.

Bit 10: target_reached Wird gesetzt, wenn der Antrieb die Zielposition erreicht hat.

Bit 11: internal_limit_active Dieses Bit drückt aus, dass eine Bewegung begrenzt wurde oder wird. In verschiedenen Modi führen unterschiedliche Warnungen zum Setzen des Bits. Es existieren folgende Zuordnungen:

Mode of operation	Warnungen, die Bit 11 setzen
alle	n04, n06, n07, n10, n11, n14
0x1 (PP), 0x88	n03, n08, n09, n20

4.4.2.3 Objekt 6060h: modes of operation (DS402)

Mit diesem Objekt wird die Betriebsart (Mode) eingestellt und mit dem Objekt 6061_h kann die eingestellte Betriebsart gelesen werden. Es können zwei Arten von Operationsmodi unterschieden werden:

herstellerspezifische Operationsmodi
Diese Operationsmodi sind auf die Gerätefunktionalitäten optimierte Betriebszustände.

Operationsmodi nach CANopen Antriebsprofil DS402
Diese Operationsmodi nach CANopen Antriebsprofil DS402 definiert.

Nach einem Betriebsartwechsel muss der entsprechende Sollwert neu gesetzt werden (z.B.: Mode Referenzierung _ Sollwert Referenzfahrtgeschwindigkeit). Wurde die Betriebsart Lage oder Tippbetrieb gespeichert, so ist nach einem RESET des Servoverstärkers die Betriebsart Referenzierung aktiv.

INFO **Ein Operationmode ist erst gültig, wenn er mit Objekt 6061_h gelesen werden konnte.**

⚠WARNUNG **Niemals die Betriebsart bei drehendem Motor umschalten! Der Antrieb könnte unkontrollierte Bewegungen ausführen.**
Das Umschalten der Betriebsart ist bei freigegebenem Verstärker grundsätzlich nur bei Drehzahl 0 erlaubt. Setzen Sie vor dem Umschalten den Sollwert auf 0.

Index	**6060_h**
Name	mode of operation
Objektcode	VAR
Datentyp	INTEGER8
Kategorie	mandatory
Zugriff	rw
PDO Mapping	möglich
Wertebereich	80h ... 88h, 1, 3, 4, 6, 7
Defaultwert	—

Wert (hex)	Betriebsart
80	Digitale Drehzahlregler
81	Analoger Drehzahlregler
82	Digitaler Stromregler
83	Analoger Stromregler
84	Elektronisches Getriebe
85	reserviert
86	reserviert
87	reserviert
88	Fahrsatz
1	Profile position mode
3	Profile velocity mode
4	Profile torque mode
6	Homing mode
7	Interpolated position mode

4.4.2.4 Objekt 6061h: Mode of Operation Display (DS402)

Mit diesem Objekt kann die über das Objekt 6060_h eingestellte Betriebsart gelesen werden. Ein Operationmode ist erst gültig, wenn er mit Objekt 6061_h gelesen werden konnte (s. auch Objekt 6060_h).

Index	**6061_h**
Name	mode of operation display
Objektcode	VAR
Datentyp	INTEGER8

Kategorie	mandatory
Zugriff	ro
PDO Mapping	möglich
Wertebereich	80h ... 88h, 1, 3, 4, 6, 7
Defaultwert	—

4.5 Factor Groups (fg) (DS402)

Factor Groups definieren die Dimensionen von Positions-, Geschwindigkeits- und Beschleunigungs-Sollwerten.

INFO

Zur Zeit ist die Einheitendefinition im CANopen Profil DS402 noch nicht endgültig festgelegt. Daher sollten die Objekte 6089h bis 609Eh nicht verwendet werden.
Die Antriebsparameter für die Einheitendefinition sollten wie folgt definiert werden:

PUNIT	**= 0 (counts)**
VUNIT	**= 0 (counts / s)**
ACCUNIT	**= 3 (counts / s²)**

4.5.1 Allgemeine Informationen

4.5.1.1 Faktoren

Es gibt eine Umrechnungsmöglichkeit für die physikalischen Dimensionen und Größen auf die antriebsinternen Einheiten (Inkremente). Dafür können einige Faktoren implementiert werden. Dieses Kapitel beschreibt den Einfluss dieser Faktoren auf das System, die Berechnungsvorschriften für diese Faktoren und die Daten, die dazu erforderlich sind.

4.5.1.2 Beziehung zwischen physikalischen und antriebsinternen Einheiten

Die in den factor groups festgelegten Faktoren definieren die Beziehung zwischen antriebsinternen Einheiten (Inkrementen) und physikalischen Einheiten.

Die Faktoren sind das Ergebnis einer Berechnung von zwei Parametern: des dimension index und des notation index. Der dimension index beschreibt dabei die physikalische Dimension, der notation index die physikalische Einheit und den Dezimalpunkt für Werte. Diese Faktoren werden zur Normierung der physikalischen Werte verwendet.

Der notation index kann auf zwei Arten verwendet werden:

- Für eine Einheit mit dezimaler Skalierung und notation index < 64 definiert der notation index den Dezimalpunkt der Einheit.
- Für eine Einheit mit nicht-dezimaler Skalierung und notation index > 64 definiert der notation index den Subindex der physikalischen Dimension der Einheit.

4.5.2 Objekts für Positionsberechnungen

4.5.2.1 Objekt 6089h: position notation index (DS402)

Index	**6089_h**
Name	position notation index
Objektcode	VAR
Datentype	INTEGER8

Kategorie	mandatory
Zugriff	rw
PDO mapping	nicht möglich
Wertebereich	INTEGER8
Defaultwert	0

Der "position notation index" skaliert Positions-Sollwerte, deren Einheiten im "position dimension index" in SI-Einheiten definiert sind, als 10er Potenzen.

Beziehung zwischen den Objekt Werten und dem herstellerspezifischen Parameter PUNIT:

Wert von Objekt6089h	ASCII Parameter PUNIT	Skalierung
FF_h	1	10^{-1}
FE_h	2	10^{-2}
FD_h	3	10^{-3}
FC_h	4	10^{-4}
FB_h	5	10^{-5}
FA_h	6	10^{-6}
$F9_h$	7	10^{-7}
$F8_h$	8	10^{-8}
$F7_h$	9	10^{-9}
0	0	1

4.5.2.2 Objekt 608Ah: position dimension index (DS402)

Index	**$608A_h$**
Name	position dimension index
Objektcode	VAR
Datentype	UNSIGNED8

Kategorie	mandatory
Zugriff	rw
PDO mapping	nicht möglich
Wertebereich	UNSIGNED8
Defaultwert	0

Der "position dimension index" definiert die SI Einheiten der verwendeten Positions Sollwerte.

Beziehung zwischen den Objekt Werten und dem herstellerspezifischen Parameter PUNIT:

Wert von Objekt608Ah	ASCII Parameter PUNIT	SI Einheit
1	9...1	m
0	0	Manufacturer specific increments

Der Parameter PUNIT kann im Servoverstärker gespeichert werden. Die Werte von Objekt 6089h und 608Ah werden von diesem Parameter initialisiert.

4.5.2.3 Objekt 608Fh: Position encoder resolution (DS402)

Dieser Index definiert das Verhältnis der Encoder Inkremente pro Umdrehung der Motorwelle. Das Objekt wird ebenso für Objekt6090 (velocity encoder resolution) verwendet.

$$\textit{position encoder resolution} = \frac{\textit{encoder increments}}{\textit{motor revolutions}}$$

Index	**608F$_h$**
Name	Position encoder resolution
Objektcode	ARRAY
Datentyp	UNSIGNED 32
Kategorie	optional

Subindex	**0**
Name	Anzahl der Einträge
Datentyp	UNSIGNED8
Kategorie	mandatory
Zugriff	ro
PDO Mapping	nicht möglich
Wertebereich	2
Defaultwert	2

Subindex	**1**
Name	Encoder increments
Kategorie	mandatory
Zugriff	rw
PDO Mapping	möglich
Wertebereich	UNSIGNED 32
Defaultwert	2^{20}

Subindex	**2**
Name	Motor revolutions
Kategorie	mandatory
Zugriff	rw
PDO Mapping	möglich
Wertebereich	UNSIGNED 32
Defaultwert	1

4.5.2.4 Objekt 6091h: Gear ratio (DS402)

Der Index definiert das Verhältnis von Umdrehung der Motorwelle (Antriebswelle) zu Umdrehung der Abtriebswelle.

$$gear\ ratio = \frac{motor\ shaft\ revolutions}{driving\ shaft\ revolutions}$$

Index	**6091_h**
Name	Gear ratio
Objektcode	ARRAY
Datentyp	UNSIGNED 32
Kategorie	optional

Subindex	**0**
Name	Anzahl der Einträge
Datentyp	UNSIGNED8
Kategorie	mandatory
Zugriff	ro
PDO Mapping	nicht möglich
Wertebereich	2
Defaultwert	2

Subindex	**1**
Name	Motor revolutions
Kategorie	mandatory
Zugriff	rw
PDO Mapping	möglich
Wertebereich	UNSIGNED 32
Defaultwert	1

Subindex	**2**
Name	Shaft revolutions
Kategorie	mandatory
Zugriff	rw
PDO Mapping	möglich
Wertebereich	UNSIGNED 32
Defaultwert	1

4.5.2.5 Objekt 6092h: Feed constant (DS402)

Hier wird das Verhältnis des Vorschubs (feed, in Positionseinheiten) pro Umdrehung der Abtriebswelle definiert. Dies schließt ein vorhandenes Getriebe mit ein.

$$feed\ constant = \frac{feed}{driving\ shaft\ revolutions}$$

Index	**6092h**
Name	Feed constant
Objektcode	ARRAY
Datentyp	UNSIGNED 32
Kategorie	optional

Subindex	**0**
Name	Anzahl der Einträge
Datentyp	UNSIGNED8
Kategorie	mandatory
Zugriff	ro
PDO Mapping	nicht möglich
Wertebereich	2
Defaultwert	2

Subindex	**1**
Name	Feed
Kategorie	mandatory
Zugriff	rw
PDO Mapping	möglich
Wertebereich	UNSIGNED 32
Defaultwert	1

Subindex	**2**
Name	Shaft revolutions
Kategorie	mandatory
Zugriff	rw
PDO Mapping	möglich
Wertebereich	UNSIGNED 32
Defaultwert	1

4.5.2.6 Objekt 6093h: Position factor (DS402)

Der Positionsfaktor konvertiert die Sollposition (in Positionseinheiten) in das interne Format (Inkremente). Diese Werte werden über die Objekte 608F und 6091 berechnet.

$$position\ factor = \frac{position\ encoder\ resolution * gear\ ratio}{feed\ cons\tan t}$$

Index	**6093h**
Name	Position factor
Objektcode	ARRAY
Datentyp	UNSIGNED 32
Kategorie	optional

Subindex	**0**
Name	Anzahl der Einträge
Datentyp	UNSIGNED8
Kategorie	mandatory
Zugriff	ro
PDO Mapping	nicht möglich
Wertebereich	2
Defaultwert	2

Subindex	**1**
Name	Zähler (position encoder resolution * gear ratio)
Kategorie	mandatory
Zugriff	rw
PDO Mapping	möglich
Wertebereich	UNSIGNED 32
Defaultwert	1

Subindex	**2**
Name	Feed constant
Kategorie	mandatory
Zugriff	rw
PDO Mapping	möglich
Wertebereich	UNSIGNED 32
Defaultwert	1

4.5.3 Objekte for velocity calculations

4.5.3.1 Objekt 608Bh: velocity notation index (DS402)

Index	**608B$_h$**
Name	velocity notation index
Objektcode	VAR
Datentype	Integer8

Kategorie	mandatory
Zugriff	rw
PDO mapping	nicht möglich
Wertebereich	INTEGER8
Defaultwert	0

Der "velocity notation index" skaliert Geschwindigkeits-Sollwerte, deren Einheiten im "velocity dimension index" in SI-Einheiten definiert sind, als 10er Potenzen.

Beziehung zwischen den Objekt Werten und dem herstellerspezifischen Parameter VUNIT:

Wert von Objekt608Bh	ASCII Parameter VUNIT	Skalierung
0	0	1
0	1	1
0	5	1
0	6	1
FD$_h$	7	10^{-3}
FD$_h$	8	10^{-3}

4.5.3.2 Objekt 608Ch: velocity dimension index (DS402)

Index	**608C$_h$**
Name	velocity dimension index
Objektcode	VAR
Datentype	UNSIGNED8

Kategorie	mandatory
Zugriff	rw
PDO mapping	nicht möglich
Wertebereich	UNSIGNED8
Defaultwert	0

Der "velocity dimension index" definiert die SI Einheiten der verwendeten Geschwindigkeits Sollwerte.

Beziehung zwischen den Objekt Werten und dem herstellerspezifischen Parameter VUNIT:

Wert von Objekt608Ch	ASCII Parameter VUNIT	SI Einheit
A6$_h$	0	m/s
A4$_h$	1	Umdr./min
A6$_h$	5	m/s
A7$_h$	6	m/min
A6$_h$	7	m/s
A7$_h$	8	m/min

Der Parameter VUNIT kann im Servoverstärker gespeichert werden. Die Werte von Objekt 608Bh und 608Ch werden von diesem Parameter initialisiert.
Nur die beschriebenen Werte für VUNIT sind mit dem Profil DS402 möglich.

4.5.4 Objekte for acceleration calculations

4.5.4.1 Objekt 608Dh: acceleration notation index (DS402)

Index	**608D$_h$**
Name	acceleration notation index
Objektcode	VAR
Datentype	INTEGER8

Kategorie	mandatory
Zugriff	rw
PDO mapping	nicht möglich
Wertebereich	INTEGER8
Defaultwert	0

Der "acceleration notation index" skaliert Beschleunigungs-Sollwerte, deren Einheiten im "acceleration dimension index" in SI-Einheiten definiert sind, als 10er Potenzen.

Beziehung zwischen den Objekt Werten und dem herstellerspezifischen Parameter ACCUNIT:

Wert von Objekt608Dh	ASCII Parameter ACCUNIT	Skalierung
0	1,5	1
FA$_h$	3	10^{-6}
FD$_h$	4	10^{-3}

4.5.4.2 Objekt 608Eh: acceleration dimension index (DS402)

Index	**608E$_h$**
Name	acceleration dimension index
Objektcode	VAR
Datentype	UNSIGNED8

Kategorie	mandatory
Zugriff	rw
PDO mapping	nicht möglich
Wertebereich	UNSIGNED8
Defaultwert	AE$_h$

Der "acceleration dimension index" definiert die SI Einheiten der verwendeten Beschleunigungs Sollwerte.

Beziehung zwischen den Objekt Werten und dem herstellerspezifischen Parameter ACCUNIT:

Wert von Objekt608Eh	ASCII Parameter ACCUNIT	SI Einheit
AE$_h$	1	rad/s²
55$_h$	3, 4, 5	m/s

Der Parameter ACCUNIT kann im Servoverstärker gespeichert werden. Die Werte von Objekt 608Dh und 608Eh werden von diesem Parameter initialisiert.
Nur die beschriebenen Werte für ACCUNIT sind mit dem Profil DS402 möglich.

4.5.4.3 Objekt 6097h: Acceleration factor (DS402)

Der Beschleunigungsfaktor konvertiert die Beschleunigung (Beschleunigungseinheit / s) in das interne Format (Inkremente / s). Dieser Faktor wird z.Zt. von Objekt6093 berechnet und ist nur lesbar.

$$acceleration\ factor = \frac{velocity\ unit * velocity\ encoder\ factor}{acceleration\ unit * second}$$

Index	**6097_h**
Name	Acceleration factor
Objektcode	ARRAY
Datentyp	UNSIGNED 32
Kategorie	optional

Subindex	**0**
Name	Anzahl der Einträge
Datentyp	UNSIGNED8
Kategorie	mandatory
Zugriff	ro
PDO Mapping	nicht möglich
Wertebereich	2
Defaultwert	2

Subindex	**1**
Name	Zähler (velocity unit * velocity encoder factor)
Kategorie	mandatory
Zugriff	ro
PDO Mapping	möglich
Wertebereich	UNSIGNED 32
Defaultwert	1

Subindex	**2**
Name	Nenner (acceleration unit * second)
Kategorie	mandatory
Zugriff	ro
PDO Mapping	möglich
Wertebereich	UNSIGNED 32
Defaultwert	1

4.6 Profile Velocity Mode (pv) (DS402)

4.6.1 Allgemeine Informationen

Der Profile Velocity Mode ermöglicht die Verarbeitung von Geschwindigkeitssollwerten und den zugehörigen Beschleunigungen.

4.6.2 Objekte, die in diesem Kapitel definiert werden

Index	Objekt	Name	Type	Zugriff
$606C_h$	VAR	velocity actual value	INTEGER32	ro
$60FF_h$	VAR	target velocity	INTEGER32	rw

4.6.3 Objekte, die in anderen Kapiteln definiert werden

Index	Objekt	Name	Type	Kapitel
6040_h	VAR	controlword	INTEGER16	dc (⇒ S.55)
6041_h	VAR	statusword	UNSIGNED16	dc (⇒ S.55)
6063_h	VAR	position actual value*	INTEGER32	pc (⇒ S.72)
6083_h	VAR	profile acceleration	UNSIGNED32	pp (⇒ S.82)
6084_h	VAR	profile deceleration	UNSIGNED32	pp (⇒ S.82)
6086_h	VAR	motion profile type	INTEGER16	pp (⇒ S.82)
6094_h	ARRAY	velocity encoder factor	UNSIGNED32	fg (⇒ S.61)

4.6.4 Objektbeschreibung

4.6.4.1 Objekt 606Ch: velocity actual value (DS402)

Das Objekt "velocity actual value” repräsentiert die aktuelle Drehzahl. Die Skalierung des Wertes hängt von dem Faktor "velocity encoder resolution” (Objekt 6094_h) ab.

Index	$606C_h$
Name	velocity actual value
Objektcode	VAR
Datentyp	INTEGER32
Mode	pv
Zugriff	ro
PDO Mapping	möglich
Einheit	velocity units
Wertebereich	$(-2^{31})..(2^{31}-1)$
Defaultwert	—
EEPROM	nein

4.6.4.2 Objekt 60FFh: target velocity (DS402)

Die Solldrehzahl (target velocity) repräsentiert den Sollwert für den Rampengenerator. Die Skalierung des Wertes hängt von dem Faktor "velocity encoder resolution” (Objekt 6094_h) ab.

Index	$60FF_h$
Name	target velocity
Objektcode	VAR
Datentyp	INTEGER32
Mode	pv
Zugriff	rw
PDO Mapping	möglich
Einheit	Inkremente
Wertebereich	$(-2^{31})..(2^{31}-1)$
Defaultwert	—
EEPROM	nein

4.7 Profile Torque Mode (tq) (DS402)

4.7.1 Allgemeine Informationen

Der *profile torque* Modus ermöglicht die Verarbeitung von Drehmomentsollwerten und den zugehörigen Strömen.

4.7.2 Objekte, die in diesem Kapitel definiert werden

Index	Objekt	Name	Type	Zugriff
6071h	VAR	Target torque	INTEGER16	rw
6073h	VAR	Max current	UNSIGNED16	rw
6077h	VAR	Torque actual value	INTEGER16	ro

4.7.3 Objektbeschreibung

4.7.3.1 Objekt 6071h: Target torque (DS402)

Dieser Parameter ist der Eingangswert für den Drehmomentregler im profile torque Modus. Die Skalierung ist 1/1000 des Nenndrehmomentes.

Index	**6071h**
Name	Target torque
Objektcode	VAR
Datentyp	INTEGER16
Kategorie	conditional; mandatory wenn tq unterstützt wird

Zugriff	rw
PDO mapping	möglich
Wertebereich	INTEGER16
Defaultwert	0

4.7.3.2 Objekt 6073h: Max current (DS402)

Dieser Parameter repräsentiert den max. erlaubten, Drehmoment erzeugenden Strom im Motor. Die Skalierung ist 1/1000 des Nennstroms.

Index	**6073h**
Name	Max current
Objektcode	VAR
Datentyp	UNSIGNED16
Kategorie	optional

Zugriff	rw
PDO mapping	möglich
Wertebereich	UNSIGNED16
Defaultwert	0

4.7.3.3 Objekt 6077h: Torque actual value (DS402)

Das aktuelle Drehmoment bezieht sich auf das augenblickliche Drehmoment im Motor. Die Skalierung ist 1/1000 des Nenndrehmoments.

Index	**6077h**
Name	Torque actual value
Objektcode	VAR
Datentyp	INTEGER16
Kategorie	optional

Zugriff	ro
PDO mapping	möglich
Wertebereich	INTEGER16
Defaultwert	0

4.8 Position Control Function (pc) (DS402)

4.8.1 Allgemeine Informationen

In diesem Kapitel werden die Positionsistwerte beschrieben, die im Zusammenhang mit dem Lageregler des Antriebs stehen. Sie finden Verwendung im Profile Position Mode.

4.8.2 Objekte, die in diesem Kapitel definiert werden

Index	Objekt	Name	Type	Zugriff
6063_h	VAR	position actual value*	INTEGER32	ro
6064_h	VAR	position actual value	INTEGER32	ro
6065_h	VAR	following error window	UNSIGNED32	rw
6067_h	VAR	position window	UNSIGNED32	rw
6068_h	VAR	position window time	UNSIGNED16	rw

4.8.3 Objekte, die in anderen Kapiteln definiert werden

Index	Objekt	Name	Type	Kapitel
$607A_h$	VAR	target position	INTEGER32	pp (⇒ S.82)
$607C_h$	VAR	home-offset	INTEGER32	hm (⇒ S.79)
607Dh	ARRAY	software position limit	INTEGER32	pp (⇒ S.82)
607Fh	VAR	max. profile velocity	UNSIGNED32	pp (⇒ S.82)
6093_h	VAR	position factor	UNSIGNED32	fg (⇒ S.61)
6094_h	ARRAY	velocity encoder factor	UNSIGNED32	fg (⇒ S.61)
6096_h	ARRAY	acceleration factor	UNSIGNED32	fg (⇒ S.61)
6040_h	VAR	controlword	INTEGER16	dc (⇒ S.55)
6041_h	VAR	statusword	UNSIGNED16	dc (⇒ S.55)

4.8.4 Objektbeschreibung

4.8.4.1 Objekt 6063h: position actual value* (DS402)

Das Objekt Aktueller Positionswert* liefert die aktuelle Istposition in Inkrementen. Die Auflösung pro Umdrehung wird über Objekt 608F in Zweierpotenzen vorgegeben. (s. ASCII-Kommando PRBASE).

Index	**6063_h**
Name	position actual value
Objektcode	VAR
Datentyp	INTEGER32
Mode	pc, pp
Zugriff	rw
PDO Mapping	möglich
Einheit	Inkremente (1 Umdr. = 2^{PRBASE})
Wertebereich	$(-2^{31})..(2^{31}-1)$
Defaultwert	2^{20}
EEPROM	nein

4.8.4.2 Objekt 6064h: position actual value (DS402)

Das Objekt Aktueller Positionswert liefert die aktuelle Istposition. Die Auflösung kann mit den Getriebefaktoren des Lagereglers geändert werden (Objekt 6092).

Index	**6064_h**
Name	position actual value
Objektcode	VAR
Datentyp	INTEGER32
Mode	pc, pp
Zugriff	rw
PDO Mapping	möglich
Einheit	position units
Wertebereich	$(-2^{31})..(2^{31}-1)$
Defaultwert	—
EEPROM	nein

4.8.4.3 Objekt 6065h: Following error window

Das Schleppfehlerfenster definiert eine tolerierte Positionswerte symmetrisch zum Positionsollwert. Ein Schleppfehler kann bei blockiertem Antrieb auftreten, bei unerreichbarer Drehzahlvorgabe oder bei fehlerhaften Reglerparametern. Wenn der eingestellte Wert des Schleppfehlerfensters 0 ist, ist die Überwachung abgeschaltet.

Index	**6065_h**
Name	Following error window
Objektcode	VAR
Datentyp	UNSIGNED32
Kategorie	optional
Zugriff	rw
PDO mapping	möglich
Wertebereich	UNSIGNED32
Defaultwert	1/4 einer Motorumdrehung

4.8.4.4 Objekt 6067h: Position window (DS402)

Das Positionsfenster definiert einen symmetrischen Bereich tolerierter Positionen relativ zur Zielposition. Wenn der aktuelle Wert des Positionsgebers im Positionsfenster liegt, wird diese Zielposition als erreicht betrachtet. Das Statuswort Bit "Target reached" wechselt auf 1.

Index	**6067_h**
Name	Position window
Objektcode	VAR
Datentyp	UNSIGNED32
Kategorie	optional
Zugriff	rw
PDO mapping	möglich
Wertebereich	UNSIGNED32
Defaultwert	4000 Positionseinheiten

4.8.4.5 Objekt 6068h: Position window time (DS402)

Wenn die aktuelle Position während der definierten Zeit (Vielfaches von 1ms) im Positionsfenster liegt, wird das zugehörige Bit 10 "Target reached" im Statuswort auf 1 gesetzt.

Index	**6068$_h$**
Name	Position window time
Objektcode	VAR
Datentyp	UNSIGNED16
Kategorie	optional

Zugriff	rw
PDO mapping	möglich
Wertebereich	UNSIGNED16
Defaultwert	—

4.8.4.6 Objekt 60F4h: Following error actual value (DS402)

Dieses Objekt liefert den aktuellen Wert des Schleppfehlers in benutzerdefinierten Einheiten.

ndex	**60F4$_h$**
Name	Following error actual value
Objektcode	VAR
Datentyp	Integer32
Kategorie	optional

Zugriff	ro
PDO mapping	möglich
Wertebereich	INTEGER32
Defaultwert	0

4.9 Interpolated Position Mode (ip) (DS402)

4.9.1 Allgemeine Informationen

Der interpolierte Positionsmodus ist auf eine einfache, geradlinige Weise realisiert. Einzelne Positionssollwerte müssen im Interpolationszeitraum übertragen werden und werden bei jedem definierten, gesendeten SYNC Telegramm übernommen. Eine lineare Interpolation wird zwischen den Sollwerten verwendet.

4.9.2 Objekte, die in diesem Kapitel definiert werden

Index	Objekt	Name	Type	Zugriff
$60C0_h$	VAR	Interpolation sub mode select	INTEG16ER	rw
$60C1_h$	ARRAY	Interpolation data record	INTEGER32	rw
$60C2_h$	RECORD	Interpolation time period	Interpolation time period	rw
$60C3_h$	ARRAY	Interpolation sync definition	UNSIGNED8	rw
$60C4_h$	RECORD	Interpolation data configuration record	Interpolation data configuration record	rw

4.9.3 Objektbeschreibung

4.9.3.1 Objekt 60C0h: Interpolation sub mode select

Im S300/S700 wird die lineare Interpolation zwischen Positionssollwerten unterstützt. Der einzige erlaubte Wert ist 0.

Index	**$60C0_h$**
Name	Interpolation sub mode select
Objektcode	VAR
Datentyp	INTEGER16
Kategorie	optional

Zugriff	rw
PDO mapping	möglich
Wertebereich	0
Defaultwert	0

4.9.3.2 Objekt 60C1h: Interpolation data record

Im S300/S700 wird nur ein einzelner Sollwert für den interpolierten Positionsmodus unterstützt. Beim linearen Interpolationsmodus kann jeder Interpolationsdatensatz (Record) als neuer Positionssollwert betrachtet werden.

Nachdem das letzte Element des Records in den Geräteeingangspuffer geschrieben wurde, wird der Zeiger des Speichers automatisch auf die nächste Speicherposition gesetzt.

Index	**$60C1_h$**
Name	Interpolation data record
Objektcode	ARRAY
Datentyp	INTEGER32
Kategorie	optional

Subindex	**0**
Beschreibung	Anzahl der Einträge
Datentyp	UNSIGNED8
Kategorie	mandatory
Zugriff	ro
PDO mapping	nicht möglich
Wertebereich	1
Defaultwert	—

Subindex	**1**
Beschreibung	x1, der erste Parameter der ip Funktion fip(x1, .. xN)
Kategorie	mandatory
Zugriff	rw
PDO mapping	möglich
Wertebereich	INTEGER32
Defaultwert	—

4.9.3.3 Objekt 60C2h: Interpolation time period

Der Interpolationszeitraum wird für den PLL (phase locked loop) synchronisierten Positionierbetrieb verwendet. Die Einheit (Subindex 1) der Zeit ist mit $10^{\text{interpolation time index}}$s gegeben, nur Vielfache von 1ms sind erlaubt. Die zwei Werte definieren den internen ASCII-Parameter PTBASE (Vielfache von 250 µs). Es müssen immer beide Werte geschrieben werden um einen neuen Interpolationszeitraum festzulegen. PTBASE wird erst dann geändert.

Index	**60C2$_h$**
Name	Interpolation time period
Objektcode	RECORD
Datentyp	Interpolation time period record (0080h)
Kategorie	optional

Subindex	**0**
Beschreibung	Anzahl der Einträge
Kategorie	mandatory
Zugriff	ro
PDO mapping	nicht möglich
Wertebereich	2
Defaultwert	2

Subindex	**1**
Beschreibung	Interpolation time units
Kategorie	mandatory
Zugriff	rw
PDO mapping	möglich
Wertebereich	UNSIGNED8
Defaultwert	1

Subindex	**2**
Beschreibung	Interpolation time index
Kategorie	mandatory
Zugriff	rw
PDO mapping	möglich
Wertebereich	-123 ... 63
Defaultwert	-3

4.9.3.4 Objekt 60C3h: Interpolation sync definition

Im S300/S700 wird das allgemein verwendete SYNC-Objekt zur Synchronisation verwendet. Daher wird nur ein fester Wert für Subindex 1 akzeptiert.

Index	**60C3$_h$**
Name	Interpolation sync definition
Objektcode	ARRAY
Datentyp	UNSIGNED8
Kategorie	optional

Subindex	**0**
Beschreibung	Anzahl der Einträge
Kategorie	mandatory
Zugriff	ro
PDO mapping	nicht möglich
Wertebereich	2
Defaultwert	1

Subindex	**1**
Beschreibung	Synchronize on group
Kategorie	mandatory
Zugriff	rw
PDO mapping	möglich
Wertebereich	UNSIGNED8
Defaultwert	0

4.9.3.5 Objekt 60C4h: Interpolation data configuration

Nur ein einzelner Positionssollwert wird im S300/S700 unterstützt. Daher ist nur der Wert 1 in Subindex 5 möglich. Alle anderen Subindizes werden auf 0 gesetzt.

Index	**60C4$_h$**
Name	Interpolation data configuration
Objektcode	RECORD
Datentyp	Interpolation data configuration record (0081h)
Kategorie	optional

Subindex	**0**
Beschreibung	Anzahl der Einträge
Kategorie	mandatory
Zugriff	ro
PDO mapping	nicht möglich
Wertebereich	6
Defaultwert	6

Subindex	**1**
Beschreibung	Maximum buffer size
Kategorie	mandatory
Zugriff	ro
PDO mapping	nicht möglich
Wertebereich	UNSIGNED32
Defaultwert	0

Subindex	**2**
Beschreibung	Actual buffer size
Kategorie	mandatory
Zugriff	rw
PDO mapping	möglich
Wertebereich	UNSIGNED32
Defaultwert	0

Subindex	**3**
Beschreibung	Buffer organization
Kategorie	mandatory
Zugriff	rw
PDO mapping	möglich
Wertebereich	UNSIGND8
Defaultwert	0

Subindex	**4**
Beschreibung	Puffer position
Kategorie	mandatory
Zugriff	rw
PDO mapping	möglich
Wertebereich	UNSIGNED16
Defaultwert	0

Subindex	**5**
Beschreibung	Size of data record
Kategorie	mandatory
Zugriff	wo
PDO mapping	möglich
Wertebereich	1...254
Defaultwert	1

Subindex	**6**
Beschreibung	Buffer clear
Kategorie	mandatory
Zugriff	wo
PDO mapping	möglich
Wertebereich	UNSIGNED8
Defaultwert	0

4.10 Homing Mode (hm) (DS402)

4.10.1 Allgemeine Informationen

Dieses Kapitel beschreibt die verschiedenen Parameter, die zur Definition einer Referenzierung benötigt werden.

4.10.2 Objekte, die in diesem Kapitel definiert werden

Index	Objekt	Name	Type	Zugriff
$607C_h$	VAR	home offset	INTEGER32	rw
6098_h	VAR	homing method	INTEGER8	rw
6099_h	ARRAY	homing speeds	UNSIGNED32	rw
$609A_h$	VAR	homing acceleration	UNSIGNED32	rw

4.10.3 Objekte, die in anderen Kapiteln definiert werden

Index	Objekt	Name	Type	Kapitel
6040_h	VAR	controlword	INTEGER16	dc (⇒ S.55)
6041_h	VAR	statusword	UNSIGNED16	dc (⇒ S.55)

4.10.4 Objektbeschreibung

4.10.4.1 Objekt 607Ch: home offset (DS402)

Der Referenzoffset ist die Differenz zwischen der Nullposition der Anwendung und des Maschinennullpunktes. Alle nachfolgenden absoluten Fahraufträge berücksichtigen den Referenzoffset.

Index	**$607C_h$**
Name	home offset
Objektcode	VAR
Datentyp	INTEGER32
Mode	hm
Zugriff	rw
PDO Mapping	möglich
Einheit	benutzerdefiniert
Wertebereich	$(-2^{31})..(2^{31}-1)$
Defaultwert	0
EEPROM	ja

4.10.4.2 Objekt 6098h: homing method (DS402)

Index	**6098_h**
Name	homing method
Objektcode	VAR
Datentyp	INTEGER8
Mode	hm
Zugriff	rw
PDO Mapping	möglich
Einheit	Positionseinheiten
Wertebereich	-128..127
Defaultwert	0
EEPROM	ja

Die folgenden Referenzfahrtarten werden unterstützt:

Methode nach DS402	Kurzbeschreibung Referenz	ASCII-Kommando
-128...-4	reserviert	—
-3	Fahren auf mechanischen Anschlag, mit Nullpunktsuche	NREF = 7
-2	Setzen des Referenzpunktes auf die aktuelle Position unter Berücksichtigung des Schleppabstandes	NREF = 6
-1	Referenzfahrt innerhalb einer Umdrehung (Drehrichtung entfernungsabhängig)	NREF = 5, DREF= 2
0	reserviert	—
1	Referenzfahrt auf negativen Endschalter, mit Nullpunktsuche, Fahrtrichtung negativ	NREF = 2, DREF= 0
2	Referenzfahrt auf positiven Endschalter, mit Nullpunktsuche, Fahrtrichtung positiv	NREF = 2, DREF= 1
3...7	werden nicht unterstützt	—
8	Referenzfahrt mit Referenzschalter, mit Nullpunktsuche, Fahrtrichtung positiv	NREF = 1, DREF= 1
9...11	werden nicht unterstützt	—
12	Referenzfahrt mit Referenzschalter, mit Nullpunktsuche, Fahrtrichtung negativ	NREF = 1, DREF= 0
13...14	werden nicht unterstützt	—
15...16	reserviert	—
17	Referenzfahrt auf negativen Endschalter, ohne Nullpunktsuche, Fahrtrichtung negativ	NREF = 4, DREF= 0
18	Referenzfahrt auf negativen Endschalter, ohne Nullpunktsuche, Fahrtrichtung positiv	NREF = 4, DREF= 1
19...23	werden nicht unterstützt	—
24	Referenzfahrt auf Referenzschalter, ohne Nullpunktsuche, Fahrtrichtung positiv	NREF = 3, DREF= 1
25...27	werden nicht unterstützt	—
28	Referenzfahrt auf Referenzschalter, ohne Nullpunktsuche, Fahrtrichtung negativ	NREF = 3, DREF= 0
29...30	werden nicht unterstützt	—
31...32	reserviert	—
33	Referenzfahrt innerhalb einer Umdrehung, Fahrtrichtung negativ	NREF = 5, DREF= 0
34	Referenzfahrt innerhalb einer Umdrehung, Fahrtrichtung positiv	NREF = 5, DREF= 1
35	Setzen des Referenzpunktes an die aktuelle Position	NREF = 0
36...127	reserviert	—

4.10.4.2.1 Beschreibung der Referenziermethoden

Durch die Auswahl einer Referenzfahrtart durch Beschreiben des Parameters homing method (Objekt 6098_h) wird folgendes bestimmt:

- das Referenzsignal (PStop, NStop, Referenzschalter)
- die Richtung der Referenzfahrt

und, wenn vorhanden

- die Position des Nullimpulses

Die Referenzposition wird durch den Referenzoffset (Objekt $607C_h$) festgelegt. Zur Anpassung der Grundeinstellung der Motorlage bei Referenzierung auf den Nullimpuls kann der herstellerspezifische Parameter ENCZERO (Objekt 3537_h, Subindex 01_h) verwendet werden.

Eine ausführliche Beschreibung der Referenzfahrtarten finden Sie in der Beschreibung der Inbetriebnahmesoftware DriveGUI.exe.

4.10.4.3 Objekt 6099h: homing speeds (DS402)

Index	**6099_h**
Name	homing speeds
Objektcode	ARRAY
Number of elements	2
Datentyp	UNSIGNED32

Subindex	**1**
Kurzbeschreibung	speed during search for switch
Mode	hm
Zugriff	rw
PDO Mapping	möglich
Einheit	Geschwindigkeitseinheit
Wertebereich	$0..(2^{32}-1)$
EEPROM	ja
Defaultwert	entspricht 60 U/min

Subindex	**2**
Kurzbeschreibung	speed during search for zero
Mode	hm
Zugriff	rw
PDO Mapping	nicht möglich
Einheit	Geschwindigkeitseinheit
Wertebereich	$0 ... (2^{32}-1)$
EEPROM	yes
Defaultwert	1/8 * Objekt 6099 sub 1

4.10.4.4 Objekt 609Ah: homing acceleration (DS402)

Index	**$609A_h$**
Name	homing acceleration
Objektcode	VAR
Datentyp	UNSIGNED32
Mode	hm
Zugriff	rw
PDO Mapping	möglich
Einheit	Beschleunigungseinheit
Wertebereich	$0..(2^{32}-1)$
Defaultwert	0
EEPROM	ja

4.10.5 Homing Mode Sequence

Mit Setzen des Bit 4 (positive Flanke) wird die Referenzfahrt gestartet. Der erfolgreiche Abschluss wird mit Bit 12 im Zustandswort angezeigt (s. Objekt 6041_h). Bit 13 zeigt einen Fehler an, der sich während der Referenzfahrt ereignet hat. Hier ist der Fehlercode auszuwerten:
Error register (Objekte 1001_h, 1003_h), manufacturer status (Objekt1002_h)

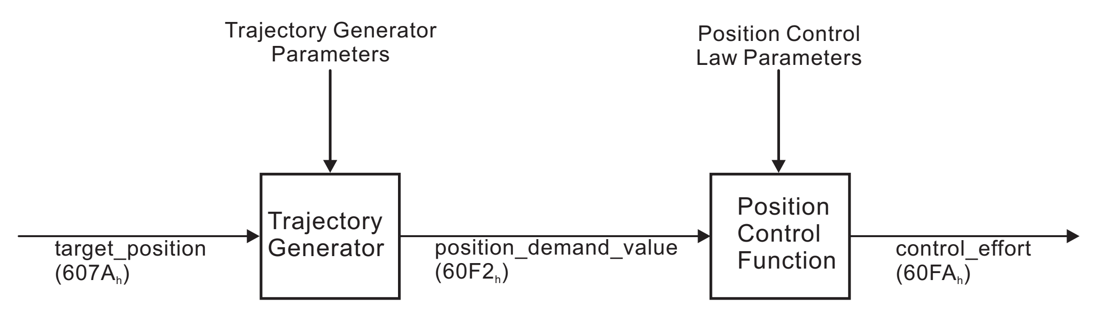
Bit 4	Bedeutung
0	Referenzfahrt inaktiv
0 ⇒ 1	Referenzfahrt starten
1	Referenzfahrt aktiv
0 ⇒ 1	Unterbrechung der Referenzfahrt

Bit 13	Bit 12	Bedeutung
0	0	Referenzpunkt nicht gesetzt bzw. Referenzfahrt noch nicht abgeschlossen
0	1	Referenzpunkt gesetzt bzw. Referenzfahrt erfolgreich abgeschlossen
1	0	Referenzfahrt konnte nicht erfolgreich abgeschlossen werden. (Schleppfehler)
1	1	kein erlaubter Zustand

4.11 Profile Position Mode (pp)

4.11.1 Allgemeine Informationen

Die Struktur dieser Betriebsart ist hier dargestellt:



Die spezielle Handshake-Verarbeitung von Controlword und Statusword wird auf Seite 83 beschrieben.

4.11.2 Objekte, die in diesem Kapitel definiert werden

Index	Objekt	Name	Type	Zugriff
$607A_h$	VAR	target position	INTEGER32	rw
$607D_h$	ARRAY	software position limit	INTEGER32	rw
607Fh	VAR	max. profile velocity	UNSIGNED32	rw
6080h	VAR	max. motor speed	UNSIGNED32	rw
6081_h	VAR	profile velocity	UNSIGNED32	rw
6083_h	VAR	profile acceleration	UNSIGNED32	rw
6084_h	VAR	profile deceleration	UNSIGNED32	rw
6085_h	VAR	quick stop deceleration	UNSIGNED32	rw
6086_h	VAR	motion profile type	INTEGER16	rw

4.11.3 Objekte, die in anderen Kapiteln definiert werden

Index	Objekt	Name	Type	Kapitel
6040_h	VAR	controlword	INTEGER16	dc (⇒ S.55)
6041_h	VAR	statusword	UNSIGNED16	dc (⇒ S.55)
6093_h	ARRAY	position factor	UNSIGNED32	fg (⇒ S.61)
6094_h	ARRAY	velocity encoder factor	UNSIGNED32	fg (⇒ S.61)
6097_h	ARRAY	acceleration factor	UNSIGNED32	fg (⇒ S.61)

4.11.4 Objektbeschreibung

4.11.4.1 Objekt 607Ah: target position (DS402)

Das Objekt target position definiert die Zielposition des Antriebes. Abhängig vom Bit 6 im controlword wird die Zielposition als relativer Weg oder als absolute Position interpretiert. Dabei kann die Art der Relativfahrt durch den herstellerspezifischen Parameter $35B9_h$ Subindex 1 weiter aufgeschlüsselt werden. Die mechanische Auflösung wird über die Getriebefaktoren Objekt 6093_h Subindex 01_h und 02_h eingestellt.

Index	**$607A_h$**
Name	target position
Objektcode	VAR
Datentyp	INTEGER32
Mode	pp
Zugriff	rw
PDO Mapping	möglich
Einheit	benutzerdefiniert
Wertebereich	$-(2^{31}-1)..(2^{31}-1)$
Defaultwert	—
EEPROM	nein

4.11.4.2 Objekt 607Dh: Software position limit (DS402)

Das Objekt Software-Endschalter beinhaltet die Subindizes *min position limit* und *max position limit*. Neue Zielpositionen werden bezogen auf diese Grenzen geprüft. Die Grenzen sind relativ zum Maschinennullpunkt, der während der Referenzfahrt einschließlich des Refernz-Offsets definiert wurde (Objekt607C).

Im Auslieferzustand sind die Software-Endschalter ausgeschaltet. Wenn die Werte geändert werden, wird im Servoverstärker eine spezielle Konfiguration durchgeführt. Daher müssen die neuen Werte gespeichert und der Verstärker neu gestartet werden um die Software-Endschalter zu aktivieren.

Index	**$607D_h$**
Name	Software position limit
Objektcode	ARRAY
Datentyp	INTEGER32
Kategorie	optional

Subindex	**0**
Beschreibung	Anzahl der Einträge
Kategorie	mandatory
Zugriff	ro
PDO mapping	nicht möglich
Wertebereich	2
Defaultwert	2

Subindex	**1**
Beschreibung	min position limit
Kategorie	mandatory
Zugriff	rw
PDO mapping	möglich
Wertebereich	INTEGER32
Defaultwert	0 (ausgeschaltet)

Subindex	**2**
Beschreibung	max position limit
Kategorie	mandatory
Zugriff	rw
PDO mapping	möglich
Wertebereich	INTEGER32
Defaultwert	0 (switched off)

4.11.4.3 Objekt 607Fh: Max profile velocity (DS402)

Die maximale Verfahrgeschwindigkeit ist die maximal zulässige Geschwindigkeit in jeder Richtung einer definierten Bewegung. Einheit analog zur Verfahrgeschwindigkeit.

Index	**607F$_h$**
Name	Max profile velocity
Objektcode	VAR
Datentyp	UNSIGNED32
Kategorie	optional

Zugriff	rw
PDO mapping	möglich
Wertebereich	UNSIGNED32
Defaultwert	entspricht der maximalen Motordrehzahl (Objekt 6080)

4.11.4.4 Objekt 6080h: Max motor speed (DS402)

Die maximale Motordrehzahl ist die maximal zulässige Drehzahl in jede Richtung und wird in rpm (U/min) angegeben. Damit wird der Motor geschützt, der korrekte Wert ist bei den Motordaten angegegeben.

Index	**6080$_h$**
Name	Max motor speed
Objektcode	VAR
Datentyp	UNSIGNED32
Kategorie	optional

Zugriff	rw
PDO mapping	möglich
Wertebereich	UNSIGNED32
Defaultwert	abhängig vom Motor

4.11.4.5 Objekt 6081h: profile velocity (DS402)

Die Profilgeschwindigkeit ist die Endgeschwindigkeit, die nach der Beschleunigungsphase eines Fahrauftrages erreicht werden soll. Die verwendete Skalierung ist abhängig von dem eingestellten "velocity encoder factor” (Objekt 6094$_h$).
Der Sollwert wird in Abhängigkeit des eingestellten Operationsmode (pp, pv) benutzt.

Index	**6081$_h$**
Name	profile velocity
Objektcode	VAR
Datentyp	UNSIGNED32
Mode	pp, pv
Zugriff	rw
PDO Mapping	möglich
Einheit	Geschwindigkeitseinheit
Wertebereich	0..(2^{32}-1)
Defaultwert	10
EEPROM	nein

4.11.4.6 Objekt 6083h: profile acceleration (DS402)

Die Beschleunigungsrampe (profile acceleration) wird in Einheiten, die der Anwender definiert hat, vorgegeben (Positionswerte / s²). Sie können mit dem Beschleunigungsfaktor (acceleration factor, definiert durch Objekt 6097 sub 1 und 2) umgewandelt werden.
Die Art der Beschleunigungsrampe kann als lineare Rampe oder als Sin² Rampe ausgewählt werden (s. Objekt 6086_h).

Index	**6083_h**
Name	profile acceleration
Objektcode	VAR
Datentyp	UNSIGNED32
Mode	pp
Zugriff	rw
PDO Mapping	möglich
Einheit	Beschleunigungseinheit
Wertebereich	0..(2^{32}-1)
Defaultwert	0

4.11.4.7 Objekt 6084h: profile deceleration (DS402)

Die Bremsrampe wird analog zur Beschleunigungsrampe behandelt. (s. Objekt 6083_h)

Index	**6084_h**
Name	profile deceleration
Objektcode	VAR
Datentyp	UNSIGNED32
Mode	pp
Zugriff	rw
PDO Mapping	möglich
Einheit	Beschleunigungseinheit
Wertebereich	0..(2^{32}-1)
Defaultwert	0

4.11.4.8 Objekt 6085h: Quick stop deceleration

Mit der Schnellstopp-Bremsrampe wird der Motor gebremst, wenn das 'Quick Stop" Kommando ausgelöst wurde.

Index	**6085_h**
Name	Quick stop deceleration
Objektcode	VAR
Datentyp	UNSIGNED32
Mode	pp
Zugriff	rw
PDO Mapping	nicht möglich
Einheit	Beschleunigungseinheit
Wertebereich	0 ... (2^{32}-1)
Defaultwert	—

4.11.4.9 Objekt 6086h: motion profile type (DS402)

Die Art der Beschleunigungsrampe kann mit diesem Objekt als lineare Rampe oder als Sin² Rampe ausgewählt werden.

Index	**6086_h**
Name	motion profile type
Objektcode	VAR
Datentyp	INTEGER16
Mode	pp
Zugriff	rw
PDO Mapping	möglich
Einheit	keine
Wertebereich	$(-2^{15})..(2^{15}-1)$
Defaultwert	—
EEPROM	ja

profile code	profile type
-32768..-1	herstellerspez. (wird nicht unterstützt)
0	linear (trapez)
1	sin²
2..32767	profilspez. Erweiterungen (wird nicht unterstützt)

4.11.4.10 Objekt 60C5h: Max acceleration

Die maximale Beschleunigung wird verwendet um Motor und Applikation zu schützen. Die Einheit ist benutzerspezifisch definiert (608Dh, 608Eh). Die Einheit wird mit dem Objekt Acceleration Factor (6097h) in Positions-Inkremente konvertiert.

Index	**$60C5_h$**
Name	Max acceleration
Objektcode	VAR
Datentyp	UNSIGNED32
Kategorie	optional
Zugriff	rw
PDO mapping	möglich
Wertebereich	UNSIGNED32
Defaultwert	—

4.11.5 Funktionelle Beschreibung

In diesem Profil werden zwei Wege der Positionssollwertübergabe an den Antrieb unterstützt.

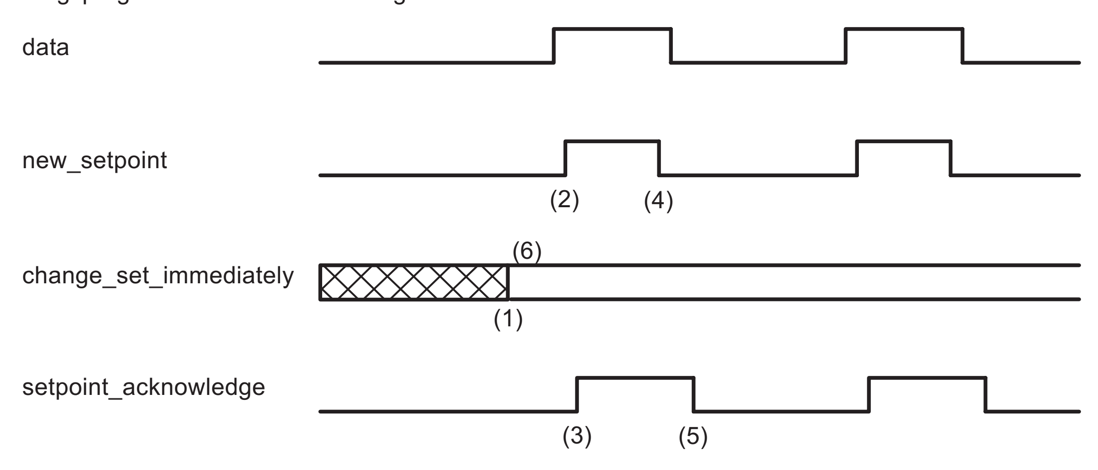
Eine Folge von Sollwerten:

Nach Erreichen der Zielposition berechnet der Antrieb sofort die Bewegung zur vorher übergebenen neuen Zielposition. Dies führt zu einer kontinuierlichen Bewegung, ohne dass der Antrieb zwischendurch auf Geschwindigkeit 0 abbremst. Beim S300/S700 ist dies nur bei Verwendung von Trapezrampen möglich.

Einzelne Sollwerte:

Nach Erreichen der Zielposition signalisiert der Antrieb an den Master, dass er das Ziel erreicht hat und erhält dann einen neuen Sollwert. Nach Erreichen der Zielposition ist die Geschwindigkeit normalerweise 0 bevor die Achse zu einer neuen Zielposition fährt.

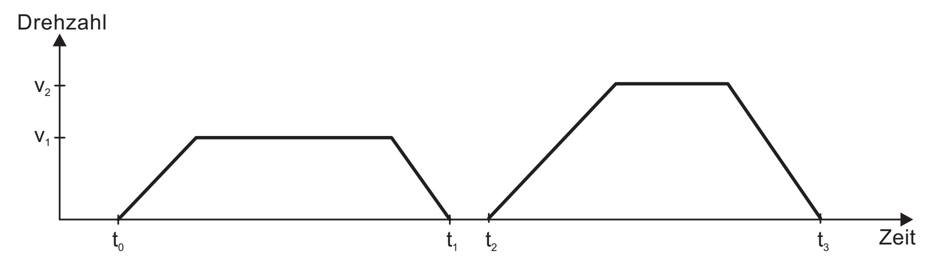
Die zwei Arten werden über das Timing der Bits "new_setpoint", "change_set_immediately" des Steuerworts und das Bit "setpoint_acknowledge" des Zustandswort gesteuert. Diese Bits ermöglichen einen Handshake - Mechanismus, der es ermöglicht einen neuen Sollwert vorzubereiten während ein alter Fahrauftrag ausgeführt wird. Das verringert Reaktionszeiten innerhalb eines Steuerungsprogramms in einer Steuerung.



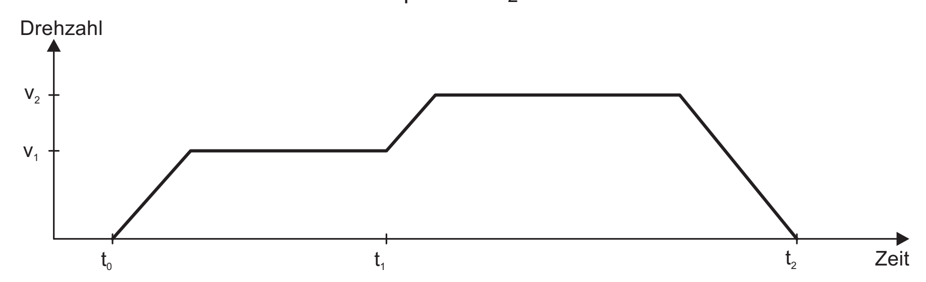
Die Bilder zeigen den Unterschied zwischen einer "Folge von Sollwerten" und einzelnen Sollwerten. Der Startwert des Bits "change_set_immediatly" im Steuerwort entscheidet über die verwendete Art. Die betrachteten Beispiele gelten nur für Trapezrampen.

Wenn das Bit **change_set_immediatly = 0** ist, wird ein einzelner Sollwert vom Antrieb erwartet (1). Nachdem Daten an den Antrieb übergeben wurden, signalisiert der Master durch den Wechsel am Bit "new_setpoint" im Steuerwort auf "1", dass die Daten gültig sind (2). Der Antrieb antwortet mit dem "setpoint_acknowledge" Bit = 1 im Zustandswort nachdem er den Wert erkannt und gespeichert hat (3). Nun kann der Master das Bit "new_setpoint" auf 0 setzen (4) woraufhin der Antrieb durch Rücksetzen des Bits "setpoint_acknowledge" signalisiert, dass er wieder neue Sollwerte entgegennehmen kann (5).

Im Bild unten führt dies zu einer Geschwindigkeit von 0, nachdem eine Rampe gefahren wurde, um die Zielposition X_1 zur Zeit t_1 zu erreichen. Nach dem Signal an den Master, dass das Ziel erreicht wurde, wird die neue Zielposition zum Zeitpunkt t_2 verarbeitet und zum Zeitpunkt t_3 erreicht.



Mit dem Bit **change_set_immediatly = 1** weist der Master den Antrieb an, direkt nach dem Erreichen des letzten Sollwerts einen neuen Sollwert zu verarbeiten. Das Timing der Signale bleibt gleich. Diese Vorgehensweise bewirkt beim Antrieb, dass er schon den nächsten Sollwert X_2 verarbeitet und Geschwindigkeit behält, wenn er die Zielposition X_1 zum Zeitpunkt t_1 erreicht. Danach fährt der Antrieb zur schon berechneten Zielposition X_2.



Bits im control word:		**Bits im status word:**	
Bit 4	new_set_point (positive Flanke!)	Bit 12	setpoint_acknowledge
Bit 5	change_set_immediately	Bit 13	Schleppfehler
Bit 6	absolut/relativ		

Hinweis zur Fahrauftragsart "relativ”:

Wird das Bit 6 gesetzt, ist die Fahrauftragsart "relativ in Abhängigkeit zur letzten Zielposition oder Istposition” aktiviert. Sollten andere Relativfahrtarten gewünscht sein, müssen diese im Vorfeld mit dem ASCII Objekt O_C (Objekt 35B9 sub 1) aktiviert werden.

Hinweis zum Mode Profile Position Mode:

Funktionale Beschreibung für den Mode: Profile Position Mode
Das Antriebsprofil DS402 unterscheidet zwei Methoden Zielpositionen anzufahren. Diese beiden Methoden werden über die Bits "new_setpoint” und "change_set_immediatly” im controlword und "setpoint_acknowledge” im statusword gesteuert. Mit Hilfe dieser Bits kann ein Fahrauftrag aufgebaut werden, während ein anderer gerade ausgeführt wird (Handshake).

- **Anfahren von mehreren Zielpositionen ohne Zwischenstopp**
  Nachdem die Zielposition erreicht wurde, wird sofort die nächste Zielposition angefahren. Voraussetzung ist, dass dem Antrieb neue Sollwerte signalisiert werden. Dies wird mit einer positiven Flanke (Bit "new_setpoint”) realisiert. Das Bit "setpoint_acknowledge” im Statusword darf hierbei nicht mehr aktiv (= 1) sein (siehe auch Handshake DS402).
  Die Geschwindigkeit wird, nach Erreichen des ersten Zielpunktes, nicht auf Null reduziert.
- **Anfahren von einer einzelnen Zielpositionen**
  Der Antrieb fährt in die Zielposition, wobei die Geschwindigkeit auf Null reduziert wird.
  Das Erreichen der Zielposition wird mit dem Bit "target_reached” im statusword signalisiert.

5 Anhang

5.1 Der Objektkanal

5.1.1 Objekt > 3500h: Herstellerspezifischer Objektkanal

Das Objektverzeichnis ist ab Index 3500_h (reservierter Objektbereich 3500_h– 3900_h) um sämtliche Geräteobjekte erweitert worden, die mit bis zu 4 Byte Nutzdaten darstellbar sind. Dieser Bereich ist dynamisch erweiterbar d.h., neue Geräteparameter, die das o. g. Datenformat erfüllen, werden bei Erweiterungen in der "Kernfirmware" **automatisch** der Tabelle angehängt. Mit Objekt 3500_h (Subindex 1, read) kann die Gesamtanzahl der Objekte im Objektkanal bestimmt werden (⇒ S. 97).

INFO **Die Objekte aus dem Objektkanal (ASCII-Parameter) können nicht in ein Prozeßdatenobjekt (PDO) gemappt werden!**

Jedes Objekt in diesem Bereich wird mit Hilfe von 8 Subindizes beschrieben. Objektstruktur:

Index	**> 3500_h**
Name	Objektabhängig
Objektcode	VAR
Datentyp	RECORD

Subindex	**0**
Beschreibung	Anzahl der Einträge
Einheit	—
Zugriff	—
PDO Mapping	nicht möglich
Datentyp	UNSIGNED8
Wertebereich	$0...2^8$-1
EEPROM	—
Defaultwert	—

Subindex	**1**
Beschreibung	Lesen / Schreiben eines Parameters
Einheit	s. jeweiliges ASCII-Kommando
Zugriff	s. jeweiliges ASCII-Kommando
PDO Mapping	nicht möglich
Datentyp	s. jeweiliges ASCII-Kommando, Übertragung immer als INTEGER32
Wertebereich	s. jeweiliges ASCII-Kommando
EEPROM	s. Subindex 4
Defaultwert	s. jeweiliges ASCII-Kommando

Subindex	**2**
Beschreibung	Lesen des unteren Grenzwertes
Einheit	s. jeweiliges ASCII-Kommando
Zugriff	ro
PDO Mapping	nicht möglich
Datentyp	s. jeweiliges ASCII-Kommando
Wertebereich	s. jeweiliges ASCII-Kommando
EEPROM	—
Defaultwert	—

Subindex	**3**
Beschreibung	Lesen des oberen Grenzwertes
Einheit	s. jeweiliges ASCII-Kommando
Zugriff	ro
PDO Mapping	nicht möglich
Datentyp	s. jeweiliges ASCII-Kommando
Wertebereich	s. jeweiliges ASCII-Kommando
EEPROM	—
Defaultwert	—

Subindex	**4**
Beschreibung	Lesen des Defaultwertes
Einheit	s. jeweiliges ASCII-Kommando
Zugriff	ro
PDO Mapping	nicht möglich
Datentyp	s. jeweiliges ASCII-Kommando
Wertebereich	s. jeweiliges ASCII-Kommando
EEPROM	—
Defaultwert	—

Subindex	**5**
Beschreibung	Lesen des Parameterformats
Einheit	—
Zugriff	ro
PDO Mapping	nicht möglich
Datentyp	s. jeweiliges ASCII-Kommando
Wertebereich	s. jeweiliges ASCII-Kommando
EEPROM	—
Defaultwert	—

Beschreibung:

Mögliche Parameterformate:

0	Funktion (kein Parameter)	**7**	INTEGER32
1	Funktion (INTEGER32 Parameter)	**8**	UNSIGNED32
2	Funktion (INTEGER32 Parameter mit Wichtung 3)	**9**	INTEGER32 (Wichtung 3)
3	INTEGER8	**10**	INTEGER32 (Wichtung 3)
4	UNSIGNED8	**11**	
5	INTEGER16	**12**	UNSIGNED32
6	UNSIGNED16	**13**	UNSIGNED16

INFO **Auf Parameter mit dem Parameterformat 0 darf nur lesend zugegriffen werden!**

Subindex	**6**
Beschreibung	Lesen der Parameter-Kontrolldaten
Einheit	—
Zugriff	ro
PDO Mapping	nicht möglich
Datentyp	UNSIGNED32
Wertebereich	$0..2^{32}-1$
EEPROM	—
Defaultwert	—

Beschreibung:

0x00010000 Nach Änderung muss die Variable abgespeichert und der Regler resettiert werden.
0x00020000 Variable wird im seriellen EEPROM abgespeichert.
0x00200000 Variable ist read-only, darf nicht über den Bus geschrieben werden.

Subindex	**7 / 8**
Beschreibung	reserviert
Einheit	—
Zugriff	ro
PDO Mapping	nicht möglich
Datentyp	UNSIGNED32
Wertebereich	$0..2^{32}-1$
EEPROM	—
Defaultwert	—

5.1.2 ASCII Kommando Referenz

MLC=Mehrzeiliges Kommando

CAN Objekt Nummer	ASCII Kommando	ASCII Type	Beschreibung
3501 (hex)	ACC	Variable rw	Beschleunigungsrampe Drehzahlregelung
3502 (hex)	ACCR	Variable rw	Beschleunigungsrampe Referenzfahrt/Tippbetrieb
3503 (hex)	ACTFAULT	Variable rw	Fehler Stop Modus
3504 (hex)	ACTIVE	Variable ro	Endstufe freigegeben/gesperrt
3505 (hex)	ADDR	Variable ro	Stationsadresse
3506 (hex)	AENA	Variable rw	Initialisierungszustand der Software-Freigabe
3507 (hex)	ANCNFG	Variable rw	Konfiguration der analogen Eingänge
3508 (hex)	ANDB	Variable rw	Totband für den analogen Drehzahlsollwert
3509 (hex)	ANIN1	Variable ro	Spannung am Analog-Eingang SW1
350A (hex)	ANIN2	Variable ro	Spannung am Analog-Eingang SW2
350B (hex)	ANOFF1	Variable rw	Analogoffset für den Analogeingang analog input 1
350C (hex)	ANOFF2	Variable rw	Analogoffset für den Analogeingang SW2
350F (hex)	ANZERO1	Command	Offsetabgleich für den Analogeingang 1
3510 (hex)	ANZERO2	Command	Offsetabgleich für den Analogeingang SW2
3511 (hex)	AVZ1	Variable rw	Filter-Zeitkonstante für den Analogeingang 1
3512 (hex)	CALCHP	Command	Ermittlung der Hiperface-Parameter
3513 (hex)	CALCRK	Command	Ermittlung der Resolverparameter
3514 (hex)	CALCRP	Command	Ermittlung der Resolverphase
3515 (hex)	CBAUD	Variable rw	Übertragungsrate CAN-Bus
3518 (hex)	CLRFAULT	Command	Löschen des Verstärker-Fehlers
3519 (hex)	CLRHR	Command	Löschen des Bit 5 im Statutsregister STAT
351A (hex)	CLRORDER	Command	Löschen eines Fahrsatzes
351B (hex)	CLRWARN	Variable rw	Behandlung der Verstärker-Warnungen
351D (hex)	CONTINUE	Command	Fortsetzen des letzten Fahrauftrages
351E (hex)	CTUNE	Command	Optimierung der Stromreglerparameter
351F (hex)	CUPDATE	Command	Programm-Update über CAN-Bus
3522 (hex)	DEC	Variable rw	Bremsrampe für den Drehzahlsollwert
3523 (hex)	DECDIS	Variable rw	Drehzahl-Bremsrampe beim Sperren der Endstufe
3524 (hex)	DECR	Variable rw	Bremsrampe für Referenzfahrt/Tippbetrieb
3525 (hex)	DECSTOP	Variable rw	Bremsrampe in Nothaltsituation
3527 (hex)	DICONT	Variable ro	Geräte-Nennstrom
3528 (hex)	DIFVAR	MLC	Parameterunterschiede zu den Default-Einstellungen
3529 (hex)	DIPEAK	Variable ro	Geräte-Spitzenstrom
352B (hex)	DIS	Command	Software Disable der Endstufe
352C (hex)	DREF	Variable rw	Fahrtrichtung für die Referenzfahrt
352D (hex)	DRVSTAT	Variable ro	Geräte-Statusinformation
352E (hex)	DR_TYPE	Variable ro	Liefert die Endstufenkennung
352F (hex)	DUMP	MLC	Auflistung aller EEPROM-Variablen
3530 (hex)	EN	Command	Setzen des Software-Enable
3533 (hex)	ENCLINES	Variable rw	Auflösung eines Sincos-Gebers
3534 (hex)	ENCMODE	Variable rw	Auswahl der Encoder-Emulation
3535 (hex)	ENCOUT	Variable rw	Auflösung für die Encoder-Emulation (ROD)
3537 (hex)	ENCZERO	Variable rw	Nullimpulsoffset (ROD-Ausgabe)
3538 (hex)	EXTMUL	Variable rw	Multiplikationsfaktor für den externen Encoder
3539 (hex)	EXTPOS	Variable rw	Positionsquelle der Lageregelung
353A (hex)	EXTWD	Variable rw	Überwachungszeit für Feldbus-Befehle
353B (hex)	FBTYPE	Variable rw	Vorwahl der Kommutierungs - Rückführeinheit
353C (hex)	FILTMODE	rw	Smith Predictor
353E (hex)	GEARI	Variable rw	Eingangsfaktor "elekronisches Getriebe”
353F (hex)	GEARMODE	Variable rw	Masterschnittstelle für "elektronisches Getriebe”
3540 (hex)	GEARO	Variable rw	Ausgangsfaktor "elekronisches Getriebe”
3541 (hex)	GET	Command	Ausgabe der aufgezeichneten SCOPE-Daten
3542 (hex)	GP	Variable rw	Lageregler: Proportionalverstärkung
3545 (hex)	GPFFV	Variable rw	Lageregler: Vorsteuerung Drehzahl
3548 (hex)	GV	Variable rw	Drehzahlregler: Proportionalverstärkung
354B (hex)	GVFR	Variable rw	PI-PLUS Drehzahlistwertvorsteuerung

CAN Objekt Nummer	ASCII Kommando	ASCII Type	Beschreibung
354D (hex)	GVTN	Variable rw	Drehzahlregler: Nachstellzeit
3551 (hex)	HDUMP	MLC	Ausgabe aller SinCos Variablen
3552 (hex)	HICOFFS	Variable rw	Hiperface: Cosinus-Offset (Inkrementalspur)
3553 (hex)	HIFACT1	Variable rw	Hiperface: Sinus/Cosinus-Faktor (Inkrementalspur)
3554 (hex)	HISOFFS	Variable rw	Hiperface: Sinus-Offset (Inkrementalspur)
3556 (hex)	HSAVE	Command	Hiperface: Speichern der Parameter im Geber
3557 (hex)	HVER	Variable ro	Ausgabe der Hardware-Version
3558 (hex)	I	Variable ro	Stromistwert
355A (hex)	I2TLIM	Variable rw	I2t-Meldeschwelle
355B (hex)	ICMD	Variable ro	Stromsollwert
355D (hex)	ID	Variable ro	D-Anteil vom Strom-Istwert
355E (hex)	IDUMP	MLC	Ausgabe der Strom-Grenzwerte
3560 (hex)	IN	MLC	Anzeige der A/D-Spannungen
3561 (hex)	IN1	Variable ro	Zustand des digitalen Eingangs INPUT1
3561 (hex)	IN5_20	Variable ro	Zustand der digitalen Eingänge 5...20
3562 (hex)	IN1MODE	Variable rw	Funktion des digitalen Eingangs INPUT1
3562 (hex)	IN5_20MODE	Variable rw	Funktion der digitalen Eingänge 5...20
3563 (hex)	IN1TRIG	Variable rw	Hilfsvariable für IN1MODE
3563 (hex)	IN5_20TRIG	Variable rw	Hilfsvariable der digitalen Eingänge 5...20
3564 (hex)	IN2	Variable ro	Zustand des digitalen Eingangs INPUT2
3565 (hex)	IN2MODE	Variable rw	Funktion des digitalen Eingangs INPUT2
3566 (hex)	IN2TRIG	Variable rw	Hilfsvariable für IN2MODE
3567 (hex)	IN3	Variable ro	Zustand des digitalen Eingangs INPUT3
3568 (hex)	IN3MODE	Variable rw	Funktion des digitalen Eingangs INPUT3
3569 (hex)	IN3TRIG	Variable rw	Hilfsvariable für IN3MODE
356A (hex)	IN4	Variable ro	Zustand des digitalen Eingangs INPUT4
356B (hex)	IN4MODE	Variable rw	Funktion des digitalen Eingangs INPUT4
356C (hex)	IN4TRIG	Variable rw	Hilfsvariable für IN3MODE
356D (hex)	INPOS	Variable ro	In-Position-Meldung
356E (hex)	IPEAK	Variable rw	Spitzenstrom
3570 (hex)	IQ	Variable ro	Q-Anteil des Stromistwertes
3571 (hex)	ISCALE1	Variable rw	Skalierungsfaktor für den analogen Stromsollwert 1
3572 (hex)	ISCALE2	Variable rw	Skalierungsfaktor für den analogen Stromsollwert 2
3573 (hex)	K	Command	Software-Sperre der Endstufe
3574 (hex)	KC	Variable rw	Stromistwert-Vorsteuerung Stromregler
3575 (hex)	KEYLOCK	Variable rw	Sperre für die Tastenbedienung
3577 (hex)	ML	Variable rw	Statorinduktivität des Motors
3578 (hex)	LATCH2P16	Variable rw	Gelatchte 16 Bit-Position (positive Flanke)
3579 (hex)	LATCH2N16	Variable rw	Gelatchte 16 Bit-Position (negative Flanke)
357A (hex)	LATCH2P32	Variable rw	Gelatchte 32 Bit-Position (positive Flanke)
357B (hex)	LATCH2N32	Variable rw	Gelatchte 32 Bit-Position (negative Flanke)
357C (hex)	LATCH1P32	Variable rw	Gelatchte 32 Bit-Position (positive Flanke)
357D (hex)	LATCH1N32	Variable rw	Gelatchte 32 Bit-Position (negative Flanke)
357E (hex)	LED1	Variable rw	Anzeigestatus des LED1-Segmentes
357F (hex)	LED2	Variable rw	Anzeigestatus des LED2-Segmentes
3580 (hex)	LED3	Variable rw	Anzeigestatus des LED3-Segmentes
3581 (hex)	LEDSTAT	Variable rw	Seitennummer für das LED-Display
3582 (hex)	LIST	MLC	Auflistung aller ASCII-Kommandos
3583 (hex)	LOAD	Command	Laden der Parameter aus dem seriellen EEPROM
3584 (hex)	MAXTEMPE	Variable rw	Abschaltwert der Umgebungstemperatur
3585 (hex)	MAXTEMPH	Variable rw	Abschaltwert der Kühlkörpertemperatur
3586 (hex)	MAXTEMPM	Variable rw	Abschaltwert der Motortemperatur (Widerstand)
3587 (hex)	MBRAKE	Variable rw	Vorwahl für Motorbremse
3588 (hex)	MDBCNT	Variable ro	Anzahl der Motorsätze
3589 (hex)	MDBGET	Command	Info-Zeile für einen Motordatensatz
358A (hex)	MDBSET	Command	Vorwahl eines Motordatensatzes
358C (hex)	VLIM	Variable rw	Max. Drehzahl
358D (hex)	MH	Command	Start der Referenzfahrt
358E (hex)	MICONT	Variable rw	Motor Nennstrom
358F (hex)	MIPEAK	Variable rw	Motor Spitzenstrom
3591 (hex)	MJOG	Command	Start des Tippbetriebes

CAN Objekt Nummer	ASCII Kommando	ASCII Type	Beschreibung
3592 (hex)	MVANGLP	Variable rw	Drehzahlabhängige Voreilung
3593 (hex)	MKT	Variable rw	Motor KT
3595 (hex)	MLGC	Variable rw	Relative Stromreglerverstärkung bei Dauerstrom
3596 (hex)	MLGD	Variable rw	Relative Stromreglerverstärkung des D-Stromreglers
3597 (hex)	MLGP	Variable rw	Relative Stromreglerverstärkung bei Spitzenstrom
3598 (hex)	MLGQ	Variable rw	Absolute Verstärkung des Stromreglers
3599 (hex)	MNUMBER	Variable rw	Laden eines Motor-Datensatzes
359C (hex)	MPHASE	Variable rw	Phasenlage des Feedback-Systems zum Motor
359D (hex)	MPOLES	Variable rw	Anzahl der Motorpole
35A0 (hex)	MRESBW	Variable rw	Resolver-Bandbreite
35A1 (hex)	MRESPOLES	Variable rw	Anzahl der Resolverpole
35A2 (hex)	MSG	Variable rw	RS232-Ausgabe der Warnungen/Fehlermeldungen
35A3 (hex)	MSPEED	Variable rw	Motor Maximaldrehzahl
35A5 (hex)	MTANGLP	Variable rw	Stromvoreilung
35A6 (hex)	MTYPE	Variable rw	Art des Motors
35A7 (hex)	MVANGLB	Variable rw	Drehzahlabhängige Voreilung (Einsatz Phi)
35A8 (hex)	MVANGLF	Variable rw	Drehzahlabhängige Voreilung (Endwert Phi)
35A9 (hex)	M_RESET	Command	Neuübersetzung der Macro-Programme
35AA (hex)	NONBTB	Variable rw	Netz-BTB-Überprüfung ein/aus
35AD (hex)	NREF	Variable rw	Referenzfahrtart
35AE (hex)	O1	Variable rw	Zustand des digitalen Ausgangs 1
35AE (hex)	O3_18	Variable rw	Zustand des digitalen Ausgangs 1
35AF (hex)	O1MODE	Variable rw	Funktion des digitalen Ausgangs 1
35AF (hex)	O3_18MODE	Variable rw	Funktion des digitalen Ausgangs 1
35B0 (hex)	O1TRIG	Variable rw	Hilfsvariable für O1MODE
35B0 (hex)	O3_18TRIG	Variable rw	Hilfsvariable für O1MODE
35B1 (hex)	O2	Variable rw	Zustand des digitalen Ausgangs 1
35B2 (hex)	O2MODE	Variable rw	Funktion des digitalen Ausgangs 2
35B3 (hex)	O2TRIG	Variable rw	Hilfsvariable für O2MODE
35B4 (hex)	OPMODE	Variable rw	Betriebsart des Verstärkers
35B5 (hex)	OPTION	Variable ro	Slotkarten-Kennung
35B6 (hex)	OVRIDE	Variable rw	Override-Funktion
35B7 (hex)	O_ACC	Variable rw	Beschleunigungszeit für den Fahrsatz 0
35B9 (hex)	O_C	Variable rw	Steuervariable für den Fahrsatz 0
35BA (hex)	O_DEC	Variable rw	Bremszeit für den Fahrsatz 0
35BC (hex)	O_FN	Variable rw	Folgefahrsatz-Nummer für den Fahrsatz 0
35BD (hex)	O_FT	Variable rw	Verzögerungszeit für den Folge-Fahrsatz
35BE (hex)	O_P	Variable rw	Zielposition/Verfahrstrecke für den Fahrsatz 0
35BF (hex)	O_V	Variable rw	Zielgeschwindigkeit für den Fahrsatz 0
35C0 (hex)	PBAL	Variable ro	Istwert der Ballastleistung
35C1 (hex)	PBALMAX	Variable rw	Maximale Ballastleistung
35C2 (hex)	PBALRES	Variable rw	Vorwahl des Ballastwiderstandes
35C3 (hex)	PBAUD	Variable ro	Profibus-Baudrate
35C5 (hex)	PE	Variable ro	Istwert des Schleppfehlers
35C6 (hex)	PEINPOS	Variable rw	In-Position-Fenster
35C7 (hex)	PEMAX	Variable rw	Max. Schleppfehler
35C8 (hex)	PFB	Variable ro	aktuelle Lagereglerposition
35C9 (hex)	PFB0	Variable ro	Lagereglerposition über den externen Encoder
35CA (hex)	PGEARI	Variable rw	Lageregler-Auflösung (Zähler)
35CB (hex)	PGEARO	Variable rw	Lageregler-Auflösung (Nenner)
35CC (hex)	PIOBUF	Variable rw	Profibus-Daten
35CD (hex)	PMODE	Variable rw	Netz-Phase Modus
35CE (hex)	PNOID	Variable ro	Profibus-Kennung (ID)
35CF (hex)	POSCNFG	Variable rw	Achsentyp
35D0 (hex)	PPOTYP	Variable rw	Profibus PPO-Typ
35D1 (hex)	PRBASE	Variable rw	interne Lage-Auflösung
35D2 (hex)	PRD	Variable ro	20 Bit Feedback-Position
35D3 (hex)	PROMPT	Variable rw	Vorwahl des RS232-Protokolls
35D4 (hex)	PSTATE	Variable ro	Profibus-Status
35D6 (hex)	PTMIN	Variable rw	Min. Beschleunigungsrampe
35D7 (hex)	PV	Variable ro	Ist-Geschwindigkeit (Lageregler)

CAN Objekt Nummer	ASCII Kommando	ASCII Type	Beschreibung
35D8 (hex)	PVMAX	Variable rw	max. Geschwindigkeit für den Lageregler
35D9 (hex)	PVMAXN	Variable rw	max. Geschwindigkeit für den Lageregler (negativ)
35DB (hex)	PVMAXP	Variable rw	max. Geschwindigkeit für den Lageregler
35DD (hex)	READY	Variable ro	Zustand von Software-Enable
35DE (hex)	RECDONE	Variable ro	Scope: Aufnahme beendet
35DF (hex)	RECING	Variable ro	Scope: Aufzeichnug läuft
35E0 (hex)	RECOFF	Command	Scope: Abbruch einer Scope-Aufzeichnung
35E1 (hex)	RECRDY	Variable ro	Scope: Zustand der RECORD-Funktion
35E2 (hex)	REFIP	Variable rw	Spitzenstrom für die Referenzfahrt 7
35E4 (hex)	REMOTE	Variable ro	Zustand des Hardware-Enable
35E5 (hex)	RESPHASE	Variable rw	Resolverphase
35E6 (hex)	RK	Variable rw	Verstärkungsfaktor Resolver-Sinussignal
35E7 (hex)	ROFFS	Variable rw	Referenzoffset
35E8 (hex)	RS232T	Variable rw	Watch-Dog Zeit (RS232)
35E9 (hex)	RSTVAR	Command	Setzen aller Parameter auf Default-Werte
35EA (hex)	S	Command	Stop und Disable
35EB (hex)	SAVE	Command	Speichern der Daten im EEPROM
35EC (hex)	SBAUD	Variable rw	Sercos: Baudrate
35ED (hex)	SCAN	Command	Erkennung der CAN-Stationen
35EF (hex)	SERIALNO	Variable ro	Seriennummer des Verstärkers
35F0 (hex)	SETREF	Command	Setzen des Referenzpunktes
35F2 (hex)	SLEN	Variable rw	Optische Ausgangsleistung bei Sercos
35F3 (hex)	SLOTIO	Variable rw	I/O-Erweiterungskarte: IN/OUT-Zustand
35F4 (hex)	SPHAS	Variable rw	Sercos-Phase
35FA (hex)	SSTAT	Variable ro	
35FA (hex)	DUMPSLNO	Variable rw	Auflistung der numerischen EEProm-Variablen
35FB (hex)	STAT	Variable ro	Verstärker-Statuswort
35FC (hex)	STATIO	Variable ro	Status der Ein/Ausgänge
35FD (hex)	STATUS	Variable ro	detailierte Verstärker-Statusinformation
35FE (hex)	STOP	Command	Setzen des Sollwertes auf 0
35FF (hex)	STOPMODE	Variable rw	Bremsverhalten bei Disable
3600 (hex)	SWCNFG	Variable rw	Konfiguration der Software-Endschalter
3604 (hex)	SWE1	Variable rw	Softwareendschalter (kleinste Position)
3606 (hex)	SWE2	Variable rw	Softwareendschalter (größte Position)
360E (hex)	T	Command	digitaler Stromsollwert
360F (hex)	TASK	Variable ro	Task-Auslastung
3610 (hex)	TEMPE	Variable ro	Istwert der Umgebungstemperatur
3611 (hex)	TEMPH	Variable ro	Istwert der Kühlkörpertemperatur
3612 (hex)	TEMPM	Variable ro	Istwert der Motortemperatur
3613 (hex)	TRJSTAT	Variable ro	Status2-Information
3614 (hex)	TRUN	Variable ro	Betriebsstundenzähler
3617 (hex)	UVLTMODE	Variable rw	Unterspannungsmodus
3618 (hex)	V	Variable ro	Aktuelle Drehzahl
361A (hex)	VBUS	Variable ro	Zwischenkreisspannung
361B (hex)	VBUSBAL	Variable rw	Maximale Netzspannung
361C (hex)	VBUSMAX	Variable ro	Maximale Zwischenkreisspannung
361D (hex)	VBUSMIN	Variable rw	Minimale Zwischenkreisspannung
361E (hex)	VCMD	Variable ro	interner Drehzahlsollwert in UPM
3620 (hex)	VEL0	Variable rw	Stillstandsschwelle
3621 (hex)	VJOG	Variable rw	Tippbetrieb-Geschwindigkeit
3622 (hex)	VLIMP	Variable rw	Max. Drehzahl
3623 (hex)	VLIMN	Variable rw	Max. negative Drehzahl
3626 (hex)	VMUL	Variable rw	Geschwindigkeitsmultiplikator (Feldbus)
3627 (hex)	VOSPD	Variable rw	Überdrehzahl
3628 (hex)	VREF	Variable rw	Referenzfahrt-Geschwindigkeit
3629 (hex)	VSCALE1	Variable rw	SW1-Drehzahlskalierungsfaktor
362A (hex)	VSCALE2	Variable rw	SW2-Drehzahlskalierungsfaktor
362B (hex)	\	Command	Anwahl der Remote Adresse
362C (hex)	DILIM	Variable rw	DPR Strombegrenzung aktivieren
362D (hex)	DENA	Variable rw	DPR Software Disable Reset Modus
362F (hex)	KTN	Variable rw	Integralanteil des Stromreglers

CAN Objekt Nummer	ASCII Kommando	ASCII Type	Beschreibung
3630 (hex)	INPT0	Variable rw	In-Position-Verzögerung
3632 (hex)	COLDSTART	Command	Hardware-Reset des Verstärkers
3636 (hex)	WPOS	Variable ro	Freigabe der schnellen Positionsregister
3637 (hex)	SRND	Variable rw	Startposition Modulo-Achse
3638 (hex)	ERND	Variable rw	Endposition Modulo-Achse
363A (hex)	BCC	Variable ro	EEPROM-Checksumme
363C (hex)	REFMODE	Variable rw	Quelle des Nullimpulse bei Referenzfahrt
363D (hex)	VLO	Variable rw	Software Resolver/Digital Wandler Vorsteuerung
363E (hex)	WMASK	Variable rw	Warnung/Fehler-Maske
363F (hex)	WPOSE	Variable ro	Freigabe der schnellen Positionsregister 1...16
3640 (hex)	WPOSP	Variable rw	Polarität der schnellen Positionsregister 1...16
3641 (hex)	WPOSX	Variable rw	Modus der schnellen Positionsregister 1...16
3642 (hex)	MOVE	Command	Starten eines Fahrsatzes
3643 (hex)	POSRSTAT	Variable rw	Status der schnellen Positionsregister 1...16
3644 (hex)	P1		schnelle Positionsschwelle
3645 (hex)	P2		schnelle Positionsschwelle
3646 (hex)	P3		schnelle Positionsschwelle
3647 (hex)	P4		schnelle Positionsschwelle
3648 (hex)	P5		schnelle Positionsschwelle
3649 (hex)	P6		schnelle Positionsschwelle
364A (hex)	P7		schnelle Positionsschwelle
364B (hex)	P8		schnelle Positionsschwelle
364C (hex)	P9		schnelle Positionsschwelle
364D (hex)	P10		schnelle Positionsschwelle
364E (hex)	P11		schnelle Positionsschwelle
364F (hex)	P12		schnelle Positionsschwelle
3650 (hex)	P13		schnelle Positionsschwelle
3651 (hex)	P14		schnelle Positionsschwelle
3652 (hex)	P15		schnelle Positionsschwelle
3653 (hex)	P16		schnelle Positionsschwelle
3654 (hex)	PTARGET	Variable rw	letzte Fahrsatzzielposition
3655 (hex)	ACTRS232	Variable rw	Freigabe des RS232-Watchdogs
3656 (hex)	ROFFSABS	Variable rw	Referenzoffset
3657 (hex)	FW	Variable ro	Liefert die Versionsnummer der Firmware
3658 (hex)	DPRILIMIT	Variable rw	Digitale Begrenzung des Spitzenstroms über DPR
3659 (hex)	ACCUNIT	Variable rw	Art der Beschleunigungsvorgabe im System
365A (hex)	VCOMM	Variable rw	Drehzahlschwelle für Kommutierungsüberwachung
365B (hex)	MTMUX	Variable rw	Voreinstellung für zu bearbeitenden Fahrsatz
365D (hex)	REFLS	Variable rw	
365F (hex)	VUNIT	Variable rw	Systemweite Definition der Drehzahl / Geschw.
3660 (hex)	PUNIT	Variable rw	Vorgabe der Positionsauflösung
366E (hex)	TBRAKE	Variable rw	Disableverzögerungszeit bei Bremsenbetrieb
366F (hex)	TBRAKE0	Variable rw	Bremsen Lüftzeit
3670 (hex)	CMDDLY	Variable rw	Kommandoverzögerungszeit der RS232
3671 (hex)	MSLBRAKE	Variable rw	Bremsrampe bei sensorlosem Nothalt
3672 (hex)	DRVCNFG	Variable rw	Konfigurationsvariable für CAN-Bus Kompatibilität
3673 (hex)	DISDPR	Variable rw	Disablen des DPR-Zugriffs für Schreibbefehle
3675 (hex)	ESPEED	Variable r	Maximale Drehzahl in Abhängigkeit vom Gebertyp
367F (hex)	LATCH1P16	Variable rw	Gelatchte 16 Bit-Position (positive Flanke)
3680 (hex)	LATCH1N16	Variable rw	Gelachte 16 Bit-Position (negative Flanke)
3681 (hex)	EXTLATCH	Variable rw	Einstellung der Quellen für die Latcheingänge
3682 (hex)	STAGECODE	Variable r	Endstufenkennung
3683 (hex)	SYNCSRC	Variable rw	
3686 (hex)	MRS	Variable rw	Wicklungswiderstand des Stators Phase-Phase
3691 (hex)	SERCSET	Variable rw	Setze Sercos Einstellungen
3695 (hex)	SMNUMBER	Variable r	Gespeicherte Motornummer im Geber
3698 (hex)	VREF0	Variable rw	Reduzierfaktor Referenzfahrgeschwindigkeit
3699 (hex)	AN11NR	Variable rw	Nr. der INxTRIG Variable, bei analoger Vorgabe
369A (hex)	AN11RANGE	Variable rw	Bereich für die analoge Änderung von INxTRIG
36A3 (hex)	MSERIALNO	Variable rw	Motorseriennummer bei Encoder mit Parameterkanal
36A5 (hex)	VSTFR	Variable rw	Drehzahleckpunkt bei Reibungskompensation

CAN Objekt Nummer	ASCII Kommando	ASCII Type	Beschreibung
36B6 (hex)	DOVRIDE	Variable rw	Vorgabe eines digitalen Override-Faktors
36BE (hex)	INS0	Variable ro	Status von Eingang A0 der I/O Erweiterungskarte
36BF (hex)	INS1	Variable ro	Status von Eingang A1 der I/O Erweiterungskarte
36C0 (hex)	INS2	Variable ro	Status von Eingang A2 der I/O Erweiterungskarte
36C1 (hex)	INS3	Variable ro	Status von Eingang A3 der I/O Erweiterungskarte
36C2 (hex)	INS4	Variable ro	Status von Eingang A4 der I/O Erweiterungskarte
36C3 (hex)	INS5	Variable ro	Status von Eingang A5 der I/O Erweiterungskarte
36C4 (hex)	INS6	Variable ro	Status von Eingang A6 der I/O Erweiterungskarte
36C5 (hex)	INS7	Variable ro	Status von Eingang A7 der I/O Erweiterungskarte
36C6 (hex)	INS8	Variable ro	Status von FSTART_IO der I/O Erweiterungskarte
36C7 (hex)	OS1	Variable rw	Setzen von "Posreg1" der I/O Erweiterungskarte
36C8 (hex)	OS2	Variable rw	Setzen von ”Posreg2" der I/O Erweiterungskarte
36C9 (hex)	OS3	Variable rw	Setzen von "Posreg3" der I/O Erweiterungskarte
36CA (hex)	OS4	Variable rw	Setzen von ”Posreg4" der I/O Erweiterungskarte
36CB (hex)	OS5	Variable rw	Setzen von "Posreg5" der I/O Erweiterungskarte
36CE (hex)	LASTWMASK	Variable ro	Fehlerspeicher von WMASK
36D0 (hex)	WSTIME	Variable rw	Ausführungszeit der W&S - Funktion
36D1 (hex)	WSAMPL	Variable rw	Minimale Bewegung der W&S Funktion
36D2 (hex)	NREFMT	Variable rw	Referenzfahrt mit automatischem Folgefahrauftrag
36D7 (hex)	AUTOHOME	Variable rw	
36D8 (hex)	PASSCNFG	Variable rw	Passwort Funktion
36E4 (hex)	DRVCNFG2	Variable rw	Regler Zusatzfunktionen
36E5 (hex)	BUSP1	Variable rw	Zustand des Modbus+ Netzwerks
36E6 (hex)	BUSP2	Variable rw	Anzahl der Datenworte (Sollwert) bei MODBUS+
36E7 (hex)	BUSP3	Variable rw	Vorgaberichtung der Adresse bei Modbus+
36E8 (hex)	BUSP4	Variable rw	Anzahl der Datenworte (Sollwert) bei MODBUS+
36E9 (hex)	BUSP5	Variable rw	Anzahl der Datenworte (Sollwert) bei MODBUS+
36EA (hex)	BUSP6	Variable rw	Anzahl der Istwert-Datenworte über Modbus

5.1.3 Beschreibung Objektverzeichnis

Die folgende Tabelle beschreibt das Objektverzeichnis (i.V. = in Vorbereitung).

Index	Sub-index	Datentyp	Zugriff	PDO mapp.	Beschreibung	ASCII Objekt
1000h	0	UNSIGNED32	ro	—	Gerätetyp	—
1001h	0	UNSIGNED8	ro	—	Fehlerregister	—
1002h	0	UNSIGNED32	ro	yes	Herstellerspezifisches Statusregister	—
1003h		ARRAY			Vordefiniertes Fehlerfeld	—
	0	UNSIGNED8	rw	—	Anzahl der Fehler	—
	1...8	UNSIGNED32	ro	—	Standard Fehlerfeld	—
1005h	0	UNSIGNED32	rw	—	COB-ID SYNC message	—
1006h	0	UNSIGNED32	rw	No	Communication cycle period	—
1008h	0	Visible String	const	—	Hersteller Gerätename	—
1009h	0	Visible String	const	—	Hersteller Hardware Version	—
100Ah	0	Visible String	const	—	Hersteller Software Version	—
100Ch	0	UNSIGNED16	rw	—	Überwachungszeit	—
100Dh	0	UNSIGNED8	rw	—	Lifetime Faktor	—
1010h	0	UNSIGNED8	ro	—	Anzahl der Einträge	—
	1	UNSIGNED32	rw	—	Speichern aller Parameter	SAVE
1011h	0	UNSIGNED8	ro	—	Anzahl der Einträge	—
	1	UNSIGNED32	rw	—	Auf Defaultwerte zuruecksetzen	RSTVAR
1014h	0	UNSIGNED32	rw	—	COB-ID für das Emergency Object	—
1016h		RECORD			Consumer heartbeat time	
	0	UNSIGNED8	ro	—	Anzahl der Einträge	—
	1	UNSIGNED32	rw	—	Consumer heartbeat time	—
1017h	0	UNSIGNED16	rw	—	Producer heartbeat time	—
1018h		RECORD			Identitäts Objekt	—
	0	UNSIGNED8	ro	—	Anzahl der Einträge	—
	1	UNSIGNED32	ro	—	Hersteller ID	—
	2	UNSIGNED32	ro	—	Produkt Code	—
	3	UNSIGNED32	ro	—	Revisions Nummer	—
	4	UNSIGNED32	ro	—	Seriennummer	SERIALNO
1026h		ARRAY			OS Prompt	—
	0	UNSIGNED8	ro	—	Anzahl der Einträge	—
	1	UNSIGNED8	w	—	StdIn	—
	2	UNSIGNED8	ro	—	StdOut	—
1400h		RECORD			RXPDO1 Kommunikations Parameter	—
	0	UNSIGNED8	ro	—	Anzahl der Einträge	—
	1	UNSIGNED32	rw	—	RXPDO1 COB - ID	—
	2	UNSIGNED8	rw	—	Übertragungstyp RXPDO1	—
1401h		RECORD			RXPDO2 Kommunikations Parameter	—
	0	UNSIGNED8	ro	—	Anzahl der Einträge	—
	1	UNSIGNED32	rw	—	RXPDO2 COB - ID	—
	2	UNSIGNED8	rw	—	Übertragungstyp RXPDO2	—
1402h		RECORD			RXPDO3 Kommunikations Parameter	—
	0	UNSIGNED8	ro	—	Anzahl der Einträge	—
	1	UNSIGNED32	rw	—	RXPDO3 COB - ID	—
	2	UNSIGNED8	rw	—	Übertragungstyp RXPDO3	—
1403h		RECORD			RXPDO4 Kommunikations Parameter	—
	0	UNSIGNED8	ro	—	Anzahl der Einträge	—
	1	UNSIGNED32	rw	—	RXPDO4 COB - ID	—
	2	UNSIGNED8	rw	—	Übertragungstyp RXPDO4	—
1600h		RECORD			RXPDO1 Mapping Parameter	—
	0	UNSIGNED8	ro	—	Anzahl der Einträge	—
	1...8	UNSIGNED32	rw	—	Mapping für n-tes Applikation Objekt	—
1601h		RECORD			RXPDO2 Mapping Parameter	—
	0	UNSIGNED8	ro	—	Anzahl der Einträge	—
	1...8	UNSIGNED32	rw	—	Mapping für n-tes Applikation Objekt	—

Index	Sub-index	Datentyp	Zugriff	PDO mapp.	Beschreibung	ASCII Objekt
1602h		RECORD			RXPDO3 Mapping Parameter	—
	0	UNSIGNED8	ro	—	Anzahl der Einträge	—
	1...8	UNSIGNED32	rw	—	Mapping für n-tes Applikation Objekt	—
1603h		RECORD			RXPDO4 Mapping Parameter	—
	0	UNSIGNED8	ro	—	Anzahl der Einträge	—
	1...8	UNSIGNED32	rw	—	Mapping für n-tes Applikation Objekt	—
1800h		RECORD			TXPDO1 Kommunikations Parameter	—
	0	UNSIGNED8	ro	—	Anzahl der Einträge	—
	1	UNSIGNED32	rw	—	TXPDO1 COB-ID	—
	2	UNSIGNED8	rw	—	Übertragungstyp TXPDO1	—
	3	UNSIGNED16	rw	—	Verzögerungszeit	—
	4	UNSIGNED8	const	—	reserviert	—
	5	UNSIGNED16	rw	—	Ereignis-Timer	—
1801h		RECORD			TXPDO2 Kommunikations Parameter	—
	0	UNSIGNED8	ro	—	Anzahl der Einträge	—
	1	UNSIGNED32	rw	—	TXPDO2 COB-ID	—
	2	UNSIGNED8	rw	—	Übertragungstyp TXPDO2	—
	3	UNSIGNED16	rw	—	Verzögerungszeit	—
	4	UNSIGNED8	const	—	reserviert	—
	5	UNSIGNED16	rw	—	Ereignis-Timer	—
1802h		RECORD			TXPDO3 Kommunikations Parameter	—
	0	UNSIGNED8	ro	—	Anzahl der Einträge	—
	1	UNSIGNED32	rw	—	TXPDO3 COB-ID	—
	2	UNSIGNED8	rw	—	Übertragungstyp TXPDO3	—
	3	UNSIGNED16	rw	—	Verzögerungszeit	—
	4	UNSIGNED8	const	—	reserviert	—
	5	UNSIGNED16	rw	—	Ereignis-Timer	—
1803h		RECORD			TXPDO4 Kommunikations Parameter	—
	0	UNSIGNED8	ro	—	Anzahl der Einträge	—
	1	UNSIGNED32	rw	—	TXPDO4 COB-ID	—
	2	UNSIGNED8	rw	—	Übertragungstyp TXPDO4	—
	3	UNSIGNED16	rw	—	Verzögerungszeit	—
	4	UNSIGNED8	const	—	reserviert	—
	5	UNSIGNED16	rw	—	Ereignis-Timer	—
1A00h		RECORD			Mapping Parameter TXPDO1	—
	0	UNSIGNED8	ro	—	Anzahl der Einträge	—
	1...8	UNSIGNED32	rw	—	Mapping für n-tes Applikation Objekt	—
1A01h		RECORD			Mapping Parameter TXPDO2	—
	0	UNSIGNED8	ro	—	Anzahl der Einträge	—
	1...8	UNSIGNED32	rw	—	Mapping für n-tes Applikation Objekt	—
1A02h		RECORD			Mapping Parameter TXPDO3	—
	0	UNSIGNED8	ro	—	Anzahl der Einträge	—
	1...8	UNSIGNED32	rw	—	Mapping für n-tes Applikation Objekt	—
1A03h		RECORD			Mapping Parameter TXPDO4	—
	0	UNSIGNED8	ro	—	Anzahl der Einträge	—
	1...8	UNSIGNED32	rw	—	Mapping für n-tes Applikation Objekt	—
2000h	0	UNSIGNED32	ro	yes	Herstellerspezifische Warnungen	STATCODE
2014h	0	ARRAY	ro	—	Maske TxPDO Kanal 1	—
	1	UNSIGNED32	rw	—	Maske (Byte 0..3)	—
	2	UNSIGNED32	rw	—	Maske (Byte 4..7)	—
2015h	0	ARRAY	ro	—	Maske TxPDO Kanal 2	—
	1	UNSIGNED32	rw	—	Maske (Byte 0..3)	—
	2	UNSIGNED32	rw	—	Maske (Byte 4..7)	—
2016h	0	ARRAY	ro	—	Maske TxPDO Kanal 3	—
	1	UNSIGNED32	rw	—	Maske (Byte 0..3)	—
	2	UNSIGNED32	rw	—	Maske (Byte 4..7)	—
2017h	0	ARRAY	ro	—	Maske TxPDO Kanal 4	—
	1	UNSIGNED32	rw	—	Maske (Byte 0..3)	—
	2	UNSIGNED32	rw	—	Maske (Byte 4..7)	—

Index	Sub-index	Datentyp	Zugriff	PDO mapp.	Beschreibung	ASCII Objekt
2030h		ARRAY			DP-Ram Variablen, nur schreibbar (PDO)	—
	0	UNSIGNED8	ro	-	Anzahl der Einträge	—
	1	INTEGER32	rww	yes	DP-Ram Variable 9	DPRVAR9
	2	INTEGER32	rww	yes	DP-Ram Variable 10	DPRVAR10
	3	INTEGER32	rww	yes	DP-Ram Variable 11	DPRVAR11
	4	INTEGER32	rww	yes	DP-Ram Variable 12	DPRVAR12
	5	INTEGER32	rww	yes	DP-Ram Variable 13	DPRVAR13
	6	INTEGER32	rww	yes	DP-Ram Variable 14	DPRVAR14
	7	INTEGER32	rww	yes	DP-Ram Variable 15	DPRVAR15
	8	INTEGER32	rww	yes	DP-Ram Variable 16	DPRVAR16
2040h		RECORD			Übersetzungsverhältnis elektr. Getriebe	—
	0	UNSIGNED8		—	Anzahl der Einträge	—
	1	INTEGER32	rw	yes	Eingangsfaktor für elektron. Getriebe	GEARI
	2	UNSIGNED32	rw	yes	Ausgangsfaktor für elektron. Getriebe	GEARO
2041h		RECORD			Istwerte elektronisches Getriebe	-
	0	UNSIGNED8	ro	-	Anzahl der Einträge	-
	1	INTEGER32	ro	yes	Slave-Geschwindigkeitssollwert	-
	2	INTEGER32	ro	yes	Mastergeschwindigkeit	-
	3	UNSIGNED8	ro	yes	Int. Kontrollwort für elektrisches Getriebe	-
2051h		ARRAY			Konfiguration der Positionsregister	-
	0	UNSIGNED8	ro	-	Anzahl der Einträge	-
	1	UNSIGNED32	rww	yes	Freigabe der Positionsregister	WPOSE
	2	UNSIGNED32	rww	yes	Modus der Positionsregister	WPOSX
	3	UNSIGNED32	rww	yes	Polarität der Positionsregister	WPOSP
2052h		ARRAY			Positionsregister, absolut	-
	0	UNSIGNED8	ro	-	Anzahl der Einträge	-
	1	INTEGER32	rww	yes	Positionsregister 1, absolut	P1
	2	INTEGER32	rww	yes	Positionsregister 2, absolut	P2
	3	INTEGER32	rww	yes	Positionsregister 3, absolut	P3
	4	INTEGER32	rww	yes	Positionsregister 4, absolut	P4
	5	INTEGER32	rww	yes	Positionsregister 5, absolut	P5
	6	INTEGER32	rww	yes	Positionsregister 6, absolut	P6
	7	INTEGER32	rww	yes	Positionsregister 7, absolut	P7
	8	INTEGER32	rww	yes	Positionsregister 8, absolut	P8
	9	INTEGER32	rww	yes	Positionsregister 9, absolut	P9
	10	INTEGER32	rww	yes	Positionsregister 10, absolut	P10
	11	INTEGER32	rww	yes	Positionsregister 11, absolut	P11
	12	INTEGER32	rww	yes	Positionsregister 12, absolut	P12
	13	INTEGER32	rww	yes	Positionsregister 13, absolut	P13
	14	INTEGER32	rww	yes	Positionsregister 14, absolut	P14
	15	INTEGER32	rww	yes	Positionsregister 15, absolut	P15
	16	INTEGER32	rww	yes	Positionsregister 16, absolut	P16
2053h		ARRAY			Positionsregister, relativ	-
	0	UNSIGNED8	ro	-	Anzahl der Einträge	-
	1	INTEGER32	rww	yes	Positionsregister 1, relativ	P1
	2	INTEGER32	rww	yes	Positionsregister 2, relativ	P2
	3	INTEGER32	rww	yes	Positionsregister 3, relativ	P3
	4	INTEGER32	rww	yes	Positionsregister 4, relativ	P4
	5	INTEGER32	rww	yes	Positionsregister 5, relativ	P5
	6	INTEGER32	rww	yes	Positionsregister 6, relativ	P6
	7	INTEGER32	rww	yes	Positionsregister 7, relativ	P7
	8	INTEGER32	rww	yes	Positionsregister 8, relativ	P8
	9	INTEGER32	rww	yes	Positionsregister 9, relativ	P9
	10	INTEGER32	rww	yes	Positionsregister 10, relativ	P10
	11	INTEGER32	rww	yes	Positionsregister 11, relativ	P11
	12	INTEGER32	rww	yes	Positionsregister 12, relativ	P12
	13	INTEGER32	rww	yes	Positionsregister 13, relativ	P13
	14	INTEGER32	rww	yes	Positionsregister 14, relativ	P14
	15	INTEGER32	rww	yes	Positionsregister 15, relativ	P15
	16	INTEGER32	rww	yes	Positionsregister 16, relativ	P16
2061h	0	UNSIGNED16	rww	yes	Strombegrenzung im Drehzahlmode	DPRILIMIT

Index	Sub-index	Datentyp	Zugriff	PDO mapp.	Beschreibung	ASCII Objekt
2080h	0	UNSIGNED16	rw	yes	Fahrauftrag für Profile Position Mode	—
2081h	0	UNSIGNED16	rw	yes	Anzeige des aktiven Fahrauftrages	MOVE
2082h	0	UNSIGNED32	wo	—	Kopieren von Fahrsätzen	OCOPY
2083h	0	UNSIGNED32	wo	—	Löschen der Flash-Fahrsätze	—
2090h		ARRAY			DP-Ram Variablen, nur lesbar	-
	0	UNSIGNED8	ro	-	Anzahl der Einträge	-
	1	INTEGER32	ro	yes	DP-Ram Variable 1	DPRVAR1
	2	INTEGER32	ro	yes	DP-Ram Variable 2	DPRVAR2
	3	INTEGER32	ro	yes	DP-Ram Variable 3	DPRVAR3
	4	INTEGER32	ro	yes	DP-Ram Variable 4	DPRVAR4
	5	INTEGER32	ro	yes	DP-Ram Variable 5	DPRVAR5
	6	INTEGER32	ro	yes	DP-Ram Variable 6	DPRVAR6
	7	INTEGER32	ro	yes	DP-Ram Variable 7	DPRVAR7
	8	INTEGER32	ro	yes	DP-Ram Variable 8	DPRVAR8
20A0h	0	INTEGER32	ro	yes	Latchposition 1, positive Flanke	LATCH1P32
20A1h	0	INTEGER32	ro	yes	Latchposition 1, negative Flanke	LATCH1N32
20A2h	0	INTEGER32	ro	yes	Latchposition 2, positive Flanke	LATCH2P32
20A3h	0	INTEGER32	ro	yes	Latchposition 2, negative Flanke	LATCH2N32
20A4h	0	UNSIGNED8	rww	yes	Latch-Kontrollregister	-
20B0h	0	INTEGER32	rww	yes	Triggervariable Digitaleingang 20	IN20TRIG
20B1h	0	UNSIGNED32	rww	yes	Steuerwort Digitaleingänge 5 .. 20	IN5 .. IN20
20B2h		ARRAY			Analoge Eingänge	-
	0	UNSIGNED8	ro	-	Anzahl der Einträge	-
	1	INTEGER16	ro	yes	Spannung Analogeingang 1	ANIN1
	2	INTEGER16	ro	yes	Spannung Analogeingang 1	ANIN2
2100h	0	UNSIGNED32	rww	yes	Schreib-Dummy	-
2101h	0	UNSIGNED32	ro	yes	Lese-Dummy	-
3500h		RECORD			ASCII Kommando MAXCMD	MAXCMD
	0	UNSIGNED8	ro	—	Anzahl der Einträge	—
	1	UNSIGNED32	ro	—	Wert	—
	2	UNSIGNED32	ro	—	Unterer Grenzwert	—
	3	UNSIGNED32	ro	—	Oberer Grenzwert	—
	4	UNSIGNED32	ro	—	Defaultwert	—
	5	UNSIGNED32	ro	—	Parameter Format	—
	6	UNSIGNED32	ro	—	Parameter Steuerdaten	—
	7	UNSIGNED32	ro	—	reserviert	—
	8	UNSIGNED32	ro	—	reserviert	—
3500h+MAXCMD		RECORD			Letzter Eintrag des ASCII Objektkanals	—
6040h	0	UNSIGNED16	w	yes	Control Word	—
6041h	0	UNSIGNED16	ro	yes	Status Word	—
6060h	0	INTEGER8	rw	yes	Betriebsarten	—
6061h	0	INTEGER8	ro	yes	Anzeige Betriebsarten	—
6063h	0	INTEGER32	ro	yes	Position-Istwert (Inkremente)	—
6064h	0	INTEGER32	ro	yes	Position-Istwert (Positions Einheiten)	PFB
6065h	0	UNSIGNED32	rw	—	Schleppfehlerfenster	PEMAX
6067h	0	UNSIGNED32	rw	—	"In Position" – Fenster	PEINPOS
6068h	0	UNSIGNED16	rw	—	"In Position" – Zeitfenster	INPT1
606Ch	0	INTEGER32	ro	yes	Geschwindigkeits-Istwert	—
6071h	0	INTEGER16	rw	yes	Drehmoment-Sollwert	—
6073h	0	UNSIGNED16	rw	-	Normierter Maximalstrom	MIPEAK
6077h	0	INTEGER16	ro	yes	Drehmoment-Istwert	—
607Ah	0	INTEGER32	rw	yes	Positions-Sollwert	O_P
607Ch	0	INTEGER32	rw	—	Referenz Offset	ROFFS
607Dh					Softwareendschalter	
	0	UNSIGNED8	ro	-	Anzahl der Einträge	
	1	INTEGER32	rw	-	Softwareendschalter 1	SWE1
	2	INTEGER32	rw	-	Softwareendschalter 2	SWE2
607Fh	0	UNSIGNED32	rw	—	Maximale Geschwindigkeit im PP-Mode	PVMAX
6080h	0	UNSIGNED32	rw	—	Maximale Motordrehzahl	VLIM
6081h	0	UNSIGNED32	rw	yes	Profil Geschwindigkeit	O_V

Index	Sub-index	Datentyp	Zugriff	PDO mapp.	Beschreibung	ASCII Objekt
6083h	0	UNSIGNED32	rw	yes	Profil Beschleunigung	O_ACC (pp) / ACC (pv)
6084h	0	UNSIGNED32	rw	yes	Profil Bremsen	O_DEC (pp) / DEC (pv)
6085h	0	UNSIGNED32	rw	—	Schnellhaltrampe	DECSTOP
6086h	0	INTEGER16	Rww	yes	Rampenprofil (Trapez / sin^2)	O_C
6089h	0	INTEGER8	rw	—	Position Notation Index	—
608Ah	0	UNSIGNED8	rw	—	Position Dimension Index	—
608Bh	0	INTEGER8	rw	—	Geschwindigkeit Notation Index	—
608Ch	0	UNSIGNED8	rw	—	Geschwindigkeit Dimension Index	—
608Dh	0	INTEGER8	rw	—	Beschleunigung Notation Index	—
608Eh	0	UNSIGNED8	rw	—	Beschleunigung Dimension Index	—
608Fh		ARRAY			Positions-Encoder Auflösung	—
	0	UNSIGNED8	ro	—	Anzahl der Einträge	—
	1	UNSIGNED32	rw	—	Encoder Inkremente	PGEARO
	2	UNSIGNED32	rw	—	Motorumdrehungen	BUSP7
6090h		ARRAY			Geschwindigkeits-Encoder Auflösung	—
	0	UNSIGNED8	ro	—	Anzahl der Einträge	—
	1	UNSIGNED32	rw	—	Encoder Incremente/s	—
	2	UNSIGNED32	rw	—	Motorumdrehungen/s	—
6091h		ARRAY			Übersetzung	—
	0	UNSIGNED8	ro	—	Anzahl der Einträge	—
	1	UNSIGNED32	rw	—	Motorumdrehungen	—
	2	UNSIGNED32	rw	—	Wellenumdrehungen	—
6092h		ARRAY			Vorschubkonstante	—
	0	UNSIGNED8	ro	—	Anzahl der Einträge	—
	1	UNSIGNED32	rw	—	Vorschub	PGEARI
	2	UNSIGNED32	rw	—	Wellenumdrehungen	—
6093h		ARRAY			Position Faktor	—
	0	UNSIGNED8	ro	—	Anzahl der Einträge	—
	1	UNSIGNED32	rw	—	Zähler	—
	2	UNSIGNED32	rw	—	Vorschubkonstante	—
6094h		ARRAY			Geschwindigkeits-Encoder Faktor	—
	0	UNSIGNED8	ro	—	Anzahl der Einträge	—
	1	UNSIGNED32	rw	—	Zähler	—
	2	UNSIGNED32	rw	—	Nenner	—
6097h		ARRAY			Beschleunigung Faktor	—
	0	UNSIGNED8	ro		Anzahl der Einträge	—
	1	UNSIGNED32	rw	—	Zähler	—
	2	UNSIGNED32	rw	—	Nenner	—
6098h	0	INTEGER8	rw	—	Referenzfahrtart	NREF, DREF
6099h		ARRAY			Geschwindigkeit Referenzfahrt	—
6099h	0	UNSIGNED8	ro	—	Anzahl der Einträge	—
	1	UNSIGNED32	rw	—	Geschwindigkeit während der Suche nach Endschaltern	VREF
	2	UNSIGNED32	rw	—	Geschwindigkeit während der Suche nach Nullmarken	VREF0
609Ah	0	UNSIGNED32	rw	—	Beschleunigung Referenzfahrt	ACCR, DECR
60C0h	0	INTEGER8	rw	—	Interpolation Sub-Mode Auswahl	—
60C1h		ARRAY			Interpolation Data Record	—
	0	UNSIGNED8	ro	—	Anzahl der Einträge	—
	1	INTEGER32	rw	yes	x1, erster Parameter der ip Funktion	—
60C2h		RECORD			Interpolation Zeitspanne	—
	0	UNSIGNED8	ro	—	Anzahl der Einträge	PTBASE
	1	UNSIGNED8	rw	—	InterpolationZeiteinheiten	—
	2	INTEGER16	rw	—	Interpolation Zeitindex	—
60C3h		ARRAY			Interpolation Sync Definition	—
	0	UNSIGNED8	ro	—	Anzahl der Einträge	—
	1	UNSIGNED8	rw	—	Synchronisiere auf Gruppe	—
	2	UNSIGNED8	rw	—	Ip Sync alle n Ereignisse	—

Index	Sub-index	Datentyp	Zugriff	PDO mapp.	Beschreibung	ASCII Objekt
60C4h		RECORD			Interpolation Daten Konfiguration	—
	0	UNSIGNED8	ro	—	Anzahl der Einträge	—
	1	UNSIGNED32	ro	—	Maximale Speichergröße	—
	2	UNSIGNED32	rw	—	Aktuelle Speichergröße	—
	3	UNSIGNED8	rw	—	Speicher Organisation	—
	4	UNSIGNED16	rw	—	Speicher Position	—
	5	UNSIGNED8	w	—	Größe eines Datensatzes	—
60C5h	0	UNSIGNED32	rw	—	Maximale Systembeschleunigung	—
60C4h	6	UNSIGNED8	w	—	Speicher leeren	—
60F4h	0	INTEGER32	ro	yes	Ist-Schleppfehler	PE
60FDh	0	UNSIGNED32	ro	yes	Digitale Eingänge	IN1 .. IN4
60FFh	0	INTEGER32	rw	yes	Geschwindigkeits-Sollwert	J
6502h	0	UNSIGNED32	ro	—	Unterstützte CANopen – Betriebsarten	—

5.2 Beispiele

Alle Beispiele gelten für den Servoverstärker S300/S700.

5.2.1 Setup

5.2.1.1 Prinzipielle Prüfung des CAN Verbindungsaufbaus

Beim Einschalten des S300/S700 wird eine Boot-up Message auf den Bus gesendet. Wenn sich im Bussystem kein geeigneter Empfänger findet, wird dieses Telegramm fortlaufend weiter gesendet. Kann ein vorhandener CAN - Master diese Nachricht nicht erkennen, überprüfen Sie die Kommunikation:

- Überprüfung des Buskabels: richtiger Wellenwiderstand, korrekte Abschlusswiderstände an beiden Enden?
- Multimeterüberprüfung des Ruhepegels der Busleitungen CAN-H und CAN-L gegenüber der CAN-GND - Leitung (ca. 2.5 V).
- Oszilloskop - Überprüfung der Ausgangssignale an CAN-H und CAN-L am S300/S700. Werden Signale auf den Bus gesendet? Spannungsdifferenz zwischen CAN-H und CAN-L bei logischer "0" ca. 2-3 V.
- Werden die Signale bei angeschlossenem Master nicht mehr weitergesendet?
- Überprüfen der Master-Hardware!
- Überprüfung der Master-Software!

5.2.1.2 Beispiel: Bedienung der Zustandsmaschine

INFO

Die Zustandsmaschine muss beim Hochfahren sequentiell bedient werden. Ein Überspringen eines Zustandes ist nicht möglich.

Nach dem Einschalten des S300/S700 und dem Erkennen der Boot-Up-Message kann die Kommunikation über SDOs aufgenommen werden, z.B. können so Parameter abgefragt oder geschrieben werden oder die Zustandsmaschine des Antriebs gesteuert werden.

Der Zustand der Zustandsmaschine kann über die Abfrage des Objekts 6041 Sub 0 erkannt werden. Direkt nach dem Einschalten erhält man dann z.B. als Antwortwert ein 0240_h. Dies entspricht dem Zustand „Switch on disabled“.

Auf dem CAN-Bus würde man als Daten folgendes sehen:

COB-ID	Control Byte	Index		Sub-index	Daten	Kommentar
		Low byte	High byte			
603	40	41	60	00_h	40 00 00 00	
583	4B	41	60	00_h	40 02 00 00	
	2 Byte Daten				Status	

Ist die Leistungsspannung vorhanden und das Hardware-Enable auf High-Signal (24 V gegen DGND) kann durch Schreiben des Controlwords (Objekt6040 Sub 0) versucht werden, den Antrieb in den Zustand „Switched on“ zu schalten. Im Erfolgsfall wird dies in der SDO - Antwort positiv quittiert (Controlbyte 0 im Datenfeld = 60_h).

Switch on

Die Nachrichten sehen dann wie folgt aus:

COB-ID	Control byte	Index		Sub-index	Daten	Kommentar
		Low byte	High byte			
603	2B	40	60	00_h	06 00 00 00	Shut down
583	60	40	60	00_h	00 00 00 00	
603	2B	40	60	00_h	07 00 00 00	Switch on
583	60	40	60	00_h	00 00 00 00	

control word = 0x0007 Bedeutung: Bit 0, Bit 1, Bit 2 set ⇒ Switch On, *Disable Voltage* off, *Quick Stop* off

Status Abfrage 2

Der neue Zustand kann dann wieder abgefragt werden und liefert das folgende Ergebnis:

COB-ID	Control byte	Index		Sub-index	Daten	Kommentar
		Low byte	High byte			
603	40	41	60	00_h	—	Status abfragen
583	4B	41	60	00_h	33 02 00 00	

Status = 0x0233 Bedeutung: Bit 0, Bit 1, Bit 5 set ⇒ ready to Switch On,
Bit 4 set ⇒ Voltage enabled
Bit 9 set ⇒ Remote, Bedienung über RS232 auch möglich

5.2.1.3 Beispiel: Tippbetrieb über SDO

Der Motor soll mit einer konstanten Drehzahl arbeiten.

COB-ID	Control byte	Index		Sub-index	Daten	Kommentar
		Low byte	High byte			
603	2F	60	60	00_h	03 00 00 00	Betriebsart “Profile Velocity”
583	60	60	60	00_h	00 00 00 00	
603	23	FF	60	00_h	00 00 00 00	Sollwert=0
583	60	FF	60	00_h	00 00 00 00	
603	2B	40	60	00_h	06 00 00 00	shutdown
583	60	40	60	00_h	00 00 00 00	
603	2B	40	60	00_h	07 00 00 00	switch on
583	60	40	60	00_h	00 00 00 00	
603	2B	40	60	00_h	0F 00 00 00	enable operation
583	60	40	60	00_h	00 00 00 00	
603	23	FF	60	00h	00 41 00 00	Sollwert=16640_{dec} / PGEARI=Umdr./sec
583	60	FF	60	00h	00 00 00 00	für PGEARI=10000, Sollwert=1,664 sec^{-1} 99,84 U/min
603	2B	40	60	00_h	0F 01 00 00	Zwischenstopp
583	60	40	60	00_h	00 00 00 00	

5.2.1.4 Beispiel: Drehmoment-Modus über SDO

Der Motor soll ein konstantes Drehmoment abgeben. In diesen Fall ist es sinnvoll die maximal zu erreichende Drehzahl des Motors über den Parameter ICMDVLIM zu begrenzen. Der Parameter kann über die Bediensoftware eingestellt werden.

Beispiel:

ICMDVLIM 300 ;Begrenzung der maximalen Drehzahl auf 300 U/min.
SAVE
COLDSTART

CAN Daten:

COB-ID	Control byte	Index		Sub-index	Daten	Kommentar
		Low byte	High byte			
603	2F	60	60	00_h	04 00 00 00	Betriebsart “Torque”
583	60	60	60	00_h	00 00 00 00	
603	2B	71	60	00_h	00 00 00 00	Sollwert=0
583	60	71	60	00_h	00 00 00 00	
603	2B	40	60	00_h	06 00 00 00	shutdown
583	60	40	60	00_h	00 00 00 00	
603	2B	40	60	00_h	07 00 00 00	switch on
583	60	40	60	00_h	00 00 00 00	
603	2B	40	60	00_h	0F 00 00 00	enable operation
583	60	40	60	00_h	00 00 00 00	
603	2B	71	60	00_h	90 01 00 00	Sollwert 400 mA
583	60	71	60	00_h	00 00 00 00	
603	2B	40	60	00_h	0F 01 00 00	Zwischenstopp
583	60	40	60	00_h	00 00 00 00	

5.2.1.5 Beispiel: Tippbetrieb über PDO

Generell ist es sinnvoll, nicht benutzte PDOs abzuschalten. Im Mode Drehzahl digital wird ein digitaler Drehzahlsollwert über RXPDO übertragen. Ist-Position und Ist-Drehzahl werden über ein SYNC-getriggertes TXPDO gelesen.

COB-ID	Control byte	Index		Sub-index	Daten	Kommentar
		Low byte	High byte			
603	2F	60	60	00_h	03 00 00 00	Betriebsart "Profile Velocity"
583	60	60	60	00_h	00 00 00 00	
603	2F	00	16	00_h	00 00 00 00	Löschen der Einträge für das erste RXPDO
583	60	00	16	00_h	00 00 00 00	
603	23	00	16	01_h	20 00 FF 60	Mapping RXPDO1, Objekt60FF,Sub-Index 0 Drehzahlsollwert, Datenlänge 32bit
583	60	00	16	01_h	00 00 00 00	
603	2F	00	16	00_h	01 00 00 00	Zahl der gemappten Objekte bestätigen
583	60	00	16	00_h	00 00 00 00	
603	2F	00	1A	00_h	00 00 00 00	Löschen der Einträge für das erste TXPDO
583	60	00	1A	00_h	00 00 00 00	
603	23	00	1A	01_h	20 00 64 60	mapping TXPDO1/1, Objekt6064, Sub-Index 0 current Positionsistwert in SI Einheiten, Datenlänge 32bit
583	60	00	1A	01_h	00 00 00 00	
603	23	00	1A	02_h	20 00 6C 60	mapping TXPDO1/2, Objekt606C,Sub-Index 0 current Drehzahlistwert, Datenlänge 32Bit
583	60	00	1A	02_h	00 00 00 00	
603	2F	00	1A	00h	02 00 00 00	Zahl der gemappten Objekte prüfen
583	60	00	1A	00h	00 00 00 00	
603	2F	00	18	02_h	01 00 00 00	Einstellen TXPDO1 auf synchron, Übertragung mit jedem SYNC
583	60	00	18	02_h	00 00 00 00	
603	23	01	18	01_h	83 02 00 80	disable TPDO2, setze Bit 31 (80h)
583	60	01	18	01_h	00 00 00 00	
603	23	02	18	01_h	83 03 00 80	disable TPDO3
583	60	02	18	01_h	00 00 00 00	
603	23	03	18	01_h	83 04 00 80	disabled TPDO4
583	60	03	18	01_h	00 00 00 00	
603	23	01	14	01_h	03 03 00 80	disabled RPDO2
583	60	01	14	01_h	00 00 00 00	
603	23	02	14	01_h	03 04 00 80	disabled RPDO3
583	60	02	14	01_h	00 00 00 00	
603	23	03	14	01_h	03 05 00 80	disabled RPDO4
583	60	03	14	01_h	00 00 00 00	
000					01 03	Freigabe NMT
603	2B	40	60	00_h	06 00 00 00	shutdown
583	60	40	60	00h	00 00 00 00	
603	2B	40	60	00_h	07 00 00 00	switch on
583	60	40	60	00_h	00 00 00 00	
603	2B	40	60	00_h	0F 00 00 00	enable operation
583	60	40	60	00_h	00 00 00 00	
203					00 40	Sollwert V= 98,3 rpm Berechnet wie folgt: 4000h=16384dec=Nsoll; 10000=PGEARI; (Nsoll/60)xPGEARI=V (16384/10000)x60=98,3 rpm
080						sende SYNC
183					FE 45 01 00 A6 AB 1A 00	Antwort, Position und Nist Pos.= 00 01 45 FE = 83454 [Si Einheiten]; Nist = (001A AB A6) / 17894,4dec = 97,7 rpm 17894,4 ist der Umrechnungsfaktor.
603	2B	40	60	00_h	0F 01 00 00	Zwischenstopp
583	60	40	60	00_h	00 00 00 00	

5.2.1.6 Beispiel: Drehmoment-Modus über PDO

Generell ist es sinnvoll nicht benutzte PDOs abzuschalten. Weiterhin soll das erste TXPDO den Stromistwert mit jedem SYNC Telegramm übertragen.

COB-ID	Control byte	Index		Sub-index	Daten	Kommentar
		Low byte	High byte			
603	2F	60	60	00_h	04 00 00 00	Betriebsart “Torque”
583	60	60	60	00_h	00 00 00 00	
603	2F	00	16	00_h	00 00 00 00	Löschen der Einträge für das erste RXPDO
583	60	00	16	00_h	00 00 00 00	
603	23	00	16	01_h	10 00 71 60	Mapping RXPDO1, Objekt6071,Sub-Index 0 Stromsollwert, Datenlänge 16 Bit
583	60	00	16	01_h	00 00 00 00	
603	2F	00	16	00_h	01 00 00 00	Zahl der gemappten Objekte prüfen
583	60	00	16	00_h	00 00 00 00	
603	2F	00	1A	00_h	00 00 00 00	Löschen der Einträge für das erste TXPDO
583	60	00	1A	00_h	00 00 00 00	
603	23	00	1A	01_h	10 00 77 60	Mapping TXPDO1, Objekt6077,Sub-Index 0 Stromistwert, Datenlänge16bit
583	60	00	1A	01_h	00 00 00 00	
603	2F	00	1A	00_h	01 00 00 00	Zahl der gemappten Objekte prüfen
583	60	00	1A	00_h	00 00 00 00	
603	2F	00	18	02_h	01 00 00 00	Einstellen TXPDO1 auf synchron, Übertragung mit jedem SYNC
583	60	00	18	02_h	00 00 00 00	
603	23	01	18	01_h	83 02 00 80	disable TPDO2, Bit 31 (80h)
583	60	01	18	01h	00 00 00 00	
603	23	02	18	01_h	83 03 00 80	disable TPDO3
583	60	02	18	01_h	00 00 00 00	
603	23	03	18	01_h	83 04 00 80	disabled TPDO4
583	60	03	18	01_h	00 00 00 00	
603	23	01	14	01_h	03 03 00 80	disabled RPDO2
583	60	01	14	01_h	00 00 00 00	
603	23	02	14	01_h	03 04 00 80	disabled RPDO3
583	60	02	14	01_h	00 00 00 00	
603	23	03	14	01_h	03 05 00 80	disabled RPDO4
583	60	03	14	01_h	00 00 00 00	
000					01 03	Freigabe NMT
603	2B	40	60	00_h	06 00 00 00	shutdown
583	60	40	60	00h	00 00 00 00	
603	2B	40	60	00_h	07 00 00 00	switch on
583	60	40	60	00_h	00 00 00 00	
603	2B	40	60	00_h	0F 00 00 00	enable operation
583	60	40	60	00_h	00 00 00 00	
203					12 02	Sollwert 530 mA
080						sende SYNC
183					19 02	Istwert 537 mA
603	2B	40	60	00_h	0F 01 00 00	Zwischenstopp
583	60	40	60	00_h	00 00 00 00	

5.2.1.7 Beispiel: Referenzfahrt über SDO

Beim Betrieb des S300/S700 als Linearachse, muss, bevor Positionierungen durchgeführt werden können, ein Referenzpunkt festgelegt werden. Dies erfolgt durch das Ausführen einer Referenzfahrt im Homing Mode (0x6).

Hier wird beispielhaft das Vorgehen im Modus *Homing* aufgezeigt.

Einige die Referenzfahrt betreffenden Parameter werden über den Bus eingestellt. Wenn man sicher sein kann, dass niemand die Parameter im Gerät geändert hat, kann dieser Teil entfallen, da die Geräte die Daten nullspannungssicher speichern können. Die Eingänge müssen als Endschalter konfiguriert sein.

Da im DS402 die Einheiten-Parameter noch nicht abschließend definiert sind, müssen Sie die folgende Einstellung wählen.

PUNIT = 0 (counts)
VUNIT = 0 (counts/s)
ACCUNIT = 3 (counts/s^2)

Die Reglergrundeinstellung haben Sie mit der Bediensoftware bereits vorgenommen. Die Auflösung ist in unserem Beispiel auf 10000 µm/Umdrehung eingestellt.

COB-ID	**Control byte**	**Index**		**Sub-index**	**Daten**	**Kommentar**
		Low byte	**High byte**			
703	00					Boot-up Meldung
603	40	41	60	00_h	00 00 00 00	Lese Profilzustand
583	4B	41	60	00_h	40 02 00 00	
603	23	99	60	01_h	10 27 00 00	v_{ref}=10000 counts/s bis zum Erreichen des Endschalters
583	60	99	60	01_h	00 00 00 00	
603	23	99	60	02_h	88 13 00 00	v_{ref}=5000 counts/s vom Endschalter zum Nullpunkt
583	60	99	60	02_h	00 00 00 00	
603	23	9A	60	00_h	10 27 00 00	Brems- und Beschl.-Rampen 1000counts/s²
583	60	9A	60	00_h	00 00 00 00	
603	23	7C	60	00_h	A8 61 00 00	Referenz-Offset 25000counts
583	60	7C	60	00_h	00 00 00 00	

Art der Referenzfahrt (6098)

COB-ID	Control byte	Index		Sub-index	Daten	Kommentar
		Low byte	High byte			
603	2F	60	60	00_h	06 00 00 00	Betriebsart = homing
583	60	60	60	00_h	00 00 00 00	
603	40	41	60	00_h	00 00 00 00	Lese Profilzustand, Antwort: 0250h
583	4B	41	60	00_h	40 02 00 00	Voltage Enabled
603	2B	40	60	00_h	06 00 00 00	Controlword Übergang_2,"ready to
583	60	40	60	00_h	00 00 00 00	switch on". Shutdown
603	2B	40	60	00_h	07 00 00 00	Übergang_3, "switch on". switch on
583	60	40	60	00_h	00 00 00 00	
603	2B	40	60	00_h	0F 00 00 00	Übergang_4,"operation enable"
583	60	40	60	00_h	00 00 00 00	
603	40	41	60	00_h	00 00 00 00	Lese Profilzustand
583	4B	41	60	00_h	37 02 00 00	
603	2B	40	60	00_h	1F 00 00 00	Homing_operation_start
583	60	40	60	00_h	00 00 00 00	
603	40	41	60	00_h	00 00 00 00	Lese Profilzustand,
583	4B	41	60	00_h	37 02 00 00	Antwort: Referenzierung nicht abgeschlossen
603	40	41	60	00_h	00 00 00 00	Lese Profilzustand,
583	4B	41	60	00_h	37 16 00 00	Antwort: Referenzierung abgeschlossen

Das Bit 12 im SDO 6041 gibt an ob die Referenzierung abgeschlossen wurde. Das zwischenzeitliche Lesen des Profilzustandes ist für den Funktionsablauf nicht zwingend notwendig, es soll nur der Verdeutlichung dienen.

5.2.1.8 Beispiel: Starte Fahrauftrag über SDO aus dem internen Speicher des S300/S700

Dieses Beispiel setzt die Definition eines Fahrauftrages z.B. mit der Inbetriebnahmesoftware voraus. Vor dem Start von absoluten Fahraufträgen muss eine Referenzfahrt durchgeführt werden.

COB-ID	Control byte	Index		Sub-index	Daten	Kommentar
		Low byte	High byte			
603	2F	60	60	00_h	01 00 00 00	Betriebsart = Lage
583	60	60	60	00_h	00 00 00 00	
603	2B	40	60	00_h	06 00 00 00	Shut Down
583	60	40	60	00_h	00 00 00 00	
603	2B	40	60	00_h	07 00 00 00	Switch On
583	60	40	60	00_h	00 00 00 00	
603	2B	40	60	00h	0F 00 00 00	Enable Operation
583	60	40	60	00h	00 00 00 00	
603	2B	80	20	00h	03 00 00 00	Anwahl Fahrsatz 3
583	60	80	20	00h	00 00 00 00	
603	2B	40	60	00h	3F 00 00 00	Start mit new SETPOINT und
583	60	40	60	00h	00 00 00 00	CHANGE_SET_IMMEDIATELY
603	40	81	20	00h	00 00 00 00	Lese gestarteten Fahrsatz
583	4B	81	20	00h	03 00 00 00	Antwort: Fahrsatz 3 wird abgearbeitet

5.2.1.9 Beispiel: Bedienung des Profile-Position Modes

Dieses Beispiel zeigt die Bedienung des *Profile position mode*. Dazu werden die PDOs wie folgt eingestellt:

Erstes RPDO

Für das erste RPDO ist kein Mapping erforderlich, da das Default Mapping das Controlword in das RXPDO1 einträgt

Zweites RPDO

COB-ID	Control byte	Index		Sub-index	Daten	Kommentar
		Low byte	High byte			
603	2F	01	16	00_h	00 00 00 01	RPDO2: Mapping löschen
583	60	01	16	00_h	00 00 00 00	
603	23	01	16	01_h	20 00 7A 60	RPDO2, Eintrag1: target_position
583	60	01	16	01_h	00 00 00 00	
603	23	01	16	02_h	20 00 81 60	RPDO2, Eintrag2: profile_velocity
583	60	01	16	02_h	00 00 00 00	
603	2F	01	16	00_h	02 00 00 00	Zahl der gemappten Objekte
583	60	01	16	00_h	00 00 00 00	

Erstes TPDO

COB-ID	Control byte	Index		Sub-index	Daten	Kommentar
		Low byte	High byte			
603	2F	00	1A	00_h	00 00 00 01	TPDO1: Mapping löschen
583	60	00	1A	00_h	00 00 00 00	
603	23	00	1A	01_h	10 00 41 60	TPDO1, Eintrag 1: profile statusword
583	60	00	1A	01_h	00 00 00 00	
603	2F	00	1A	00_h	01 00 00 00	Zahl der gemappten Objekte
583	60	00	1A	00_h	00 00 00 00	

Zweites TPDO

COB-ID	Control byte	Index		Sub-index	Daten	Kommentar
		Low byte	High byte			
603	2F	01	1A	00_h	00 00 00 01	TPDO2: Mapping löschen
583	60	01	1A	00_h	00 00 00 00	
603	23	01	1A	01_h	20 00 64 60	TPDO2, Eintrag 1: position_actual_value
583	60	01	1A	01_h	00 00 00 00	
603	23	01	1A	02_h	20 00 6C 60	TPDO2, Eintrag 2: velocity_actual_value
583	60	01	1A	02_h	00 00 00 00	
603	2F	01	1A	00_h	02 00 00 00	Zahl der gemappten Objekte
583	60	01	1A	00_h	00 00 00 00	

Das zweite TPDO soll mit jedem SYNC vom Antrieb gesendet werden.

COB-ID	Control byte	Index		Sub-index	Daten	Kommentar
		Low byte	High byte			
603	2F	01	18	02_h	01 00 00 00	TPDO2 mit jedem SYNC
583	60	01	18	02_h	00 00 00 00	

Sperren der nichtbenötigten TPDOs

COB-ID	Control byte	Index Low byte	Index High byte	Sub-index	Daten	Kommentar
603	23	02	18	01h	83 03 00 80	disable TPDO3
583	60	02	18	01h	00 00 00 00	
603	23	03	18	01h	83 04 00 80	disable TPDO4
583	60	03	18	01h	00 00 00 00	

Sperren der nichtbenötigten RPDOs

COB-ID	Control byte	Index Low byte	Index High byte	Sub-index	Daten	Kommentar
603	23	02	14	01h	03 04 00 80	disable RPDO3
583	60	02	14	01h	00 00 00 00	
603	23	03	14	01h	03 05 00 80	disable RPDO4
583	60	03	14	01h	00 00 00 00	

Vorgabe der mechanischen Auflösung über Objekt 6092h, Subindex 01h und 02h. Defaultwerte sind die antriebsspezifischen Faktoren PGEARI und PGEARO:

COB-ID	Control byte	Index Low byte	Index High byte	Sub-index	Daten	Kommentar
603	23	93	60	01h	00 00 10 00	2E20 Inkremente
583	60	93	60	01h	00 00 00 00	
603	23	93	60	02h	A0 8C 00 00	3600 Nutzereinheiten
583	60	93	60	02h	00 00 00 00	

Nach der Festlegung der PDOs können diese mit dem NMT freigegeben werden:

COB-ID		Daten	Kommentar
000		01 03	enable NMT
183		40 02	Profilstatus

Nach diesen Einstellungen kann eine Referenzfahrt eingestellt und angestoßen werden.

COB-ID	Control byte	Index Low byte	Index High byte	Sub-index	Daten	Kommentar
603	2F	60	60	00h	06 00 00 00	Betriebsart = homing
583	60	60	60	00h	00 00 00 00	
603	2F	98	60	00h	0C 00 00 00	Referenzfahrt 12, negative Fahrt-
583	60	98	60	00h	00 00 00 00	richtung (DS402)
603	23	99	60	01h	40 19 01 00	Referenzfahrtgeschw.
583	80	99	60	01h	31 00 09 06	72000 Einh./s=2s-1
603	2B	40	60	00h	06 00 00 00	Übergang_2,"ready to switch on". Shutdown
583	60	40	60	00h	00 00 00 00	
183					21 02	
603	2B	40	60	00h	07 00 00 00	Übergang_3,"switch on". Switch on
583	60	40	60	00h	00 00 00 00	
183					33 02	
603	2B	40	60	00h	0F 00 00 00	Controlword: Operation Enable
583	60	40	60	00h	00 00 00 00	
183					37 02	
603	2B	40	60	00h	1F 00 00 00	Referenzfahrt starten
583	60	40	60	00h	00 00 00 00	Antworttelegramm
183					37 06	Antwort: target reached
183					37 16	Antwort: homing attained

Beenden der Referenzfahrt über das Steuerwort 1_RPDO

COB-ID		Daten	Kommentar
203		0F 00	

Umschalten in den Profile Position Mode und Rampen für Positionierungen vorgegeben

COB-ID	Control byte	Index		Sub-index	Daten	Kommentar
		Low byte	High byte			
603	2F	60	60	00h	01 00 00 00	Betriebsart = Profile Positioning
583	60	60	60	00h	00 00 00 00	
603	23	83	60	00h	32 00 00 00	50ms Beschleunigungszeit
583	60	83	60	00h	00 00 00 00	
603	23	84	60	00h	32 00 00 00	50ms Bremszeit
583	60	84	60	00h	00 00 00 00	

Sollwert

COB-ID		Daten	Kommentar
303		20 4E 00 00	Pos 8CA0 =36000µm ; V= 20000 µm/s
080			sende SYNC
283		BB F8 FF FF	

Controlword mit „new setpoint“-Bit (bit 4) gesetzt

COB-ID		Daten	Kommentar
203		1F 00	

Warten

COB-ID		Daten	Kommentar
183		37 12	setpoint acknowledge

Controlword mit „new setpoint“-Bit (bit 4) zurückgesetzt

COB-ID		Daten	Kommentar
203		0F 00	
183		37 02	reset Setpoint acknowledge

Warten

COB-ID		Daten	Kommentar
183		37 06	Antwort: target reached
080			sende SYNC
283		92 FC FF FF	Antwort: 92 FC Position FF FF Geschwindigkeit

5.2.1.10 Beispiel: ASCII Kommunikation über SDO

In diesem Beispiel wird die P-Verstärkung des Drehzahlreglers auf 6 eingestellt. Der ASCII Befehl dazu lautet: „GV 6“.

COB-ID	Control byte	Index		Sub-index	Daten	Kommentar
		Low byte	High byte			
603	2F	26	10	01_h	47 00 00 00	sende ASCII Zeichen "G"
583	60	26	10	01_h	00 00 00 00	
603	2F	26	10	01_h	56 00 00 00	sende ASCII Zeichen "V"
583	60	26	10	01_h	00 00 00 00	
603	2F	26	10	01_h	20 00 00 00	sende ASCII Zeichen "SP"
583	60	26	10	01_h	00 00 00 00	(Leerzeichen)
603	2F	26	10	01_h	36 00 00 00	sende ASCII Zeichen "6"
583	60	26	10	01_h	00 00 00 00	
603	2F	26	10	01_h	0D 00 00 00	sende ASCII Zeichen "CR"
583	60	26	10	01_h	00 00 00 00	
603	2F	26	10	01_h	0A 00 00 00	sende ASCII Zeichen "LF"
583	60	26	10	01_h	00 00 00 00	

5.2.1.11 Test für SYNC Telegramme

Konfiguration

Ziele:
a) Zielposition und Verfahrgeschwidigkeit einem PDO (2. RPDO) zuweisen.
b) Aktuelle Position einem PDO (1. TPDO) zuweisen, generiert mit jedem zweiten SYNC.
c) Statuswort und Herstellerstatus einem PDO (2. TPDO) zuweisen, generiert mit jedem dritten SYNC.

Telegramm mit den zugehörigen Antworten:

COB-ID	Control Byte	Index		Sub-index	Daten	Kommentar
		Low byte	High byte			
603	2F	01	16	00_h	00 00 00 00	RPDO2: lösche Mapping
583	60	01	16	00_h	00 00 00 00	
603	23	01	16	01_h	20 00 7A 60	RPDO2, Eintrag 1: Zielposition
583	60	01	16	01_h	00 00 00 00	
603	23	01	16	02_h	20 00 81 60	RPDO2, Eintrag 2: Verfahrgeschwindigkeit
583	60	01	16	02_h	00 00 00 00	
603	2F	01	16	00_h	02 00 00 00	RPDO2: Anzahl der gemapp-ten Objekte
583	60	01	16	00_h	00 00 00 00	
603	2F	00	1A	00_h	00 00 00 00	TPDO1: lösche Mapping
583	60	00	1A	00_h	00 00 00 00	
603	23	00	1A	01_h	20 00 64 60	TPDO1: Eintrag 1: Ist-Position
583	60	00	1A	01_h	00 00 00 00	
603	2F	00	1A	00_h	01 00 00 00	TPDO1: Anzahl der gemappten Objekte
583	60	00	1A	00_h	00 00 00 00	
603	2F	00	18	02_h	02 00 00 00	TPDO1: sende mit jedem 2. SYNC
583	60	00	18	02_h	00 00 00 00	
603	2F	01	1A	00_h	00 00 00 00	TPDO2: lösche Mapping
583	60	01	1A	00_h	00 00 00 00	
603	23	01	1A	01_h	10 00 41 60	TPDO2: Eintrag 1: Statuswort
583	60	01	1A	01_h	00 00 00 00	
603	23	01	1A	02_h	20 00 02 10	TPDO2: Eintrag 2: Herstellerstatus
583	60	01	1A	02_h	00 00 00 00	
603	2F	01	16	00_h	02 00 00 00	TPDO2: Anzahl der gemappten Objekte
583	60	01	16	00_h	00 00 00 00	
603	2F	01	18	02_h	03 00 00 00	TPDO2: sende mit jedem 3. SYNC
583	60	01	18	02_h	00 00 00 00	

SYNC Objekt

COB-ID	Kommentar
080	Objekt 181 (TPDO 1) erscheint bei jedem 2. SYNC, Objekt 281 (TPDO 2) erscheint bei jedem 3. SYNC.

Emergency Objekt

Wenn z.B. der Resolver Stecker nicht gesteckt ist, wird im Verstärker ein schwerwiegender Fehler ausgelöst. Ein Emergency Telegramm wird ausgelöst:

COB-ID	Emergency error		Error register		
	Low	High			
081	10	43	08	00 00 00 00	Motor Temperatur, Temperatur, herstellerspezifisch
081	00	00	88	00 00 00 00	

5.2.1.12 Anwendung: Elektrisches Getriebe

Für die Verwendung des Modes elektrisches Getriebe über die CANopen-Schnittstelle sind die folgenden Parameter für die Einstellung des Slaves des elektrischen Getriebes relevant:

GEARI	Objekt 2040 sub 1
GEARO	Objekt 2040 sub 2
GEARMODE	Objekt 353F sub 1
EXTPOS	Objekt 3539 sub 1
IN20MODE	Objekt 3723 sub 1
IN20TRIG	Objekt 3724 sub 1 oder Objekt 20B0 sub 0 (auch mappbar)
ACCR	Objekt 3502 sub 1
DECR	Objekt 3524 sub 1

ACCR und DECR können auch gemeinsam über das Objekt 609A (auch mappbar) eingestellt werden.

Die Übersetzung des elektrischen Getriebes wird über die Getriebefaktoren GEARI und GEARO eingestellt. Der Eingang der Masterimpulse wird über die Parameter GEARMODE und EXTPOS definiert.

Folgende Bits haben in der betriebsart elektrisches Getriebe eine besondere Bedeutung:
CANopen Statusword Bit 10 = 1: Slaveachse ist in Synchronisation
CANopen Controlword Bit 13 = 1: Stopp der Mitfahrt der Slaveachse

Über den virtuellen Eingang 20 kann bei folgenden Eingangsfunktionen (IN20MODE) die Art der Aufsynchronisierung und der Aussynchronisierung auf den Master verändert werden:

IN20MODE 42: Die Slaveachse wird über die Rampe ACCR aufsynchronisiert und über DECR aussynchronisert.

IN20MODE 43: Beim Start der Slaveachse wird die Masterposition gelatcht. Die Slaveachse wird dann über die Rampe ACCR auf die Mastergeschwindigkeit gebracht und anschließend die Wegdifferenz zum Master ausgeglichen. Aussynchronisiert wird über DECR. Über IN20TRIG kann zusätzlich ein Positionsoffset auf die Masterposition vergeben werden.

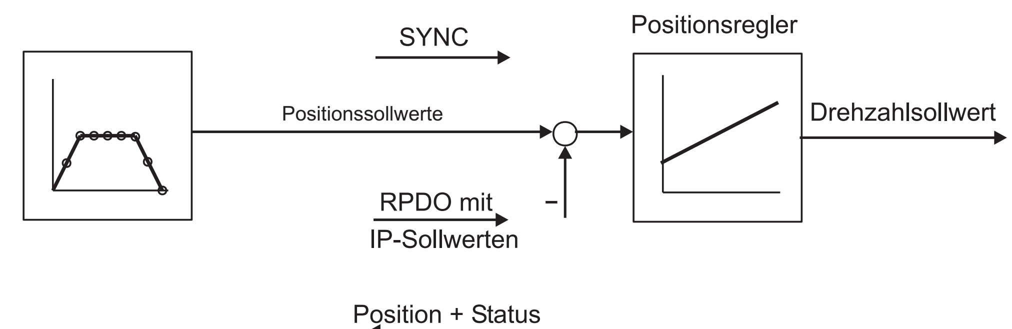
IN20MODE 51: Über IN20TRIG kann eine Wegstrecke vorgegeben werden, auf der der Slave sich einsynchronisieren soll. Aussynchronisert wird über DECR.

5.2.2 Spezielle Anwendungen

5.2.2.1 Beispiel: Externe Trajektorie mit Interpolated Position Modus

Dieses Beispiel zeigt, wie die Positions-Sollwerte für zwei Achsen mit einem PDO übertragen werden können.

Reglerstruktur der Lageregelung im Servoverstärker:



Beschreibung

Alle Daten sind hexadezimal. Im Beispiel haben die beiden Achsen im System die Stationsadressen 1 und 2.

Voraussetzungen

— Für den IP-Modus muss die interne Synchronisation verwendet werden.
  Dafür muss der Parameter SYNCSRC (Objekt 3683 sub 1) auf 3 gesetzt werden.
— Die Parameters müssen im EEPROM gespeichert werden.
— Ein Kaltstart muss durchgeführt werden, um die Synchronisation freizugeben.
— Die Achsen müssen referenziert sein.

Das gemeinsame PDO beinhaltet 2 IP (interpolierte Position) Sollwerte und kann simultan an zwei Stationen übertragen werden, wobei jede Station die für sie relevanten Daten extrahieren kann.

Die anderen Daten können mit Dummy-Daten zur Nichtbeachtung kenntlich gemacht werden (Objekt 2100 sub 0). Um dies zu erreichen, müssen beide Achsen auf dieselbe RPDO-COB-ID reagieren.

Durchführung

RPDO Mapping:

Achse 1

COB-ID	Control byte	Index		Sub-index	Daten	Kommentar
		Low byte	High byte			
601	2F	01	16	00_h	00 00 00 00	RPDO2: Lösche Mapping
581	60	01	16	00_h	00 00 00 00	
601	23	01	16	01_h	20 01 C1 60	RPDO2, Eintrag 1:
581	60	01	16	01_h	00 00 00 00	IP Sollwert Achse 1
601	23	01	16	02_h	20 00 00 21	RPDO2, Eintrag 2:
581	60	01	16	02_h	00 00 00 00	Dummy Eintrag 4 Byte
601	2F	01	16	00_h	02 00 00 00	RPDO2, Anzahl der gemappten
581	60	01	16	00_h	00 00 00 00	Objekte

Achse 2

COB-ID	Control byte	Index		Sub-index	Daten	Kommentar
		Low byte	High byte			
602	2F	01	16	00_h	00 00 00 00	RPDO2: Lösche Mapping
582	60	01	16	00_h	00 00 00 00	
602	23	01	16	02_h	20 00 00 21	RPDO2, entry 1:
582	60	01	16	02_h	00 00 00 00	Dummy Eintrag 4 bytes
602	23	01	16	01_h	20 01 C1 60	RPDO2, Eintrag 2:
582	60	01	16	01_h	00 00 00 00	IP Sollwert Achse 2
602	2F	01	16	00_h	02 00 00 00	RPDO2, Anzahl der gemappten
582	60	01	16	00_h	00 00 00 00	Objekte
602	23	01	16	01h	01 03 00 00	RPDO2: Setze COB-ID identisch
582	60	01	16	01h	00 00 00 00	zu Achse 1

Jetzt reagieren beide Achsen auf denselben COB-Identifier 0x301,
Achse 1 nimmt Bytes 0 bis 3 als IP Sollwert, Achse 2 nimmt Bytes 4 bis 7.

TPDO Mapping:

Das zweite TPDO soll die Ist-Position in Inkrementen und den Herstellerstatus beinhalten.

Achse 1

COB-ID	Control byte	Index		Sub-index	Daten	Kommentar
		Low byte	High byte			
601	2F	01	1A	00_h	00 00 00 00	TPDO2: Lösche Mapping
581	60	01	1A	00_h	00 00 00 00	
601	23	01	1A	01_h	20 00 63 60	TPDO2, Eintrag 1:
581	60	01	1A	01_h	00 00 00 00	Ist-Position in Inkrementen
601	23	01	1A	02_h	20 00 02 10	TPDO2, Eintrag 2:
581	60	01	1A	02_h	00 00 00 00	Dummy Eintrag 4 Byte
601	2F	01	1A	00_h	02 00 00 00	TPDO2, Anzahl der gemappten
581	60	01	1A	00_h	00 00 00 00	Objekte

Führen Sie dasselbe Mapping für Achse 2 durch.

Wir nehmen an, dass beide Verstärker neue Trajektorie-Werte mit jedem SYNC akzeptieren und ihre inkrementellen Positionswerte und Herstellerstati mit jedem SYNC zurückgeben. Die Kommunikationsparameter müssen entsprechend eingestellt werden:

Achse 1

COB-ID	Control byte	Index		Sub-index	Daten	Kommentar
		Low byte	High byte			
601	2F	01	14	02_h	01 00 00 00	RPDO2 Achse 1, Reaktion bei jedem SYNC
581	60	01	14	02_h	00 00 00 00	
602	2F	01	14	02_h	01 00 00 00	RPDO2 Achse 2, Reaktion bei jedem SYNC
582	60	01	14	02_h	00 00 00 00	
601	2F	01	18	02_h	01 00 00 00	TPDO2 Achse 1, Reaktion bei jedem SYNC
581	60	01	18	02_h	00 00 00 00	
602	2F	01	18	02_h	01 00 00 00	TPDO2 Achse 2, Reaktion bei jedem SYNC
582	60	01	18	02_h	00 00 00 00	

Die anderen TPDOs 3 und 4 sollten abgeschaltet werden um die Bus Belastung zu minimieren:

COB-ID	Control byte	Index		Sub-index	Daten	Kommentar
		Low byte	High byte			
601	23	02	18	01_h	81 03 00 80	TPDO3 abschalten
581	60	02	18	01_h	00 00 00 00	
601	23	03	18	01_h	81 04 00 80	TPDO4 abschalten
581	60	03	18	01_h	00 00 00 00	

Führen Sie dasselbe für Achse 2 durch.

Um Trajektorien Fahrten durchzuführen, müssen beide Servoversträker in der korrekten Betriebsart arbeiten. Dies wird eingestellt durch Index 6060_h:

COB-ID	Control byte	Index		Sub-index	Daten	Kommentar
		Low byte	High byte			
601	2F	60	60	00_h	07 00 00 00	Setze IP Modus für Achse 1
581	60	60	60	00_h	00 00 00 00	
602	2F	60	60	00_h	07 00 00 00	Setze IP Modus für Achse 2
582	60	60	60	00_h	00 00 00 00	

Die Zykluszeit für den IP-Modus soll 1 ms lang sein. Dies muss mit Objekt 60C1 sub 1 und 2 definiert werden:

COB-ID	Control byte	Index		Sub-index	Daten	Kommentar
		Low byte	High byte			
601	2F	C2	60	01_h	01 00 00 00	Interpolationszeit Einheit 1
581	60	C2	60	01_h	00 00 00 00	
601	2F	C2	60	02_h	FD 00 00 00	Interpolationszeit Index 3 -> Zykluszeit = $1 * 10^{-3}$ s
581	60	C2	60	02_h	00 00 00 00	

Führen Sie dasselbe für Achse 2 durch.

Um die Achsen zu starten, muss der Servoverstärker in den Betriebsmodus gebracht werden (operation enable) und die Netzwerk Management Funktionen müssen gestartet werden.

Die Netzwerk Management Funktionen geben die Anwendung der Process Data Objects (PDOs) und werden von folgendem Telegramm für beide Achsen initialisiert:

Schalte NMT (**N**etwork **M**anagemen**t**) Status Maschine in *operation enable*:

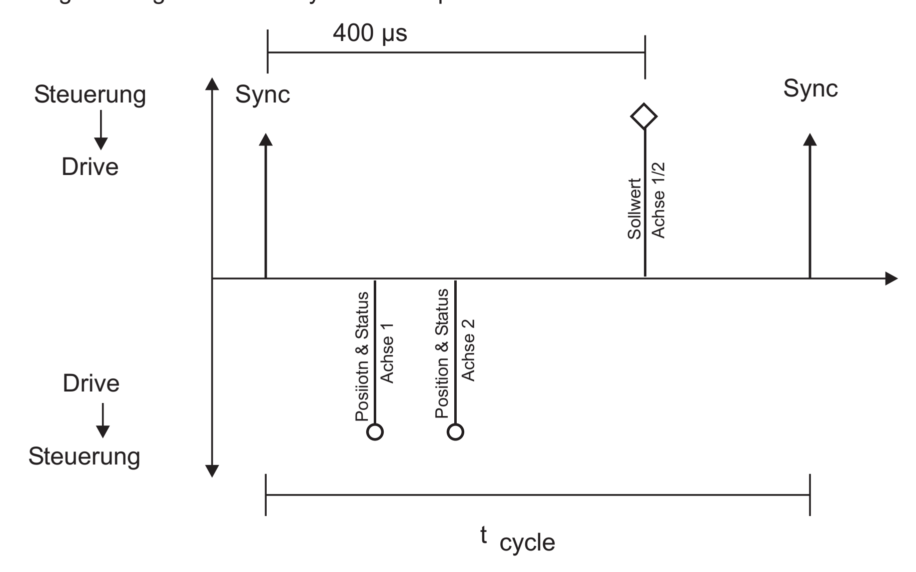
COB-ID	Command specifier (CS)	Node-ID	Kommentar
0	1	1	NMT freigegeben für alle Achsen

Als nächstes wird die Spannung für jeden Verstärker eingeschaltet und sie werden in den Betriebszustand *operation enable* geschaltet.

Dies sollte schrittweise durchgeführt werden mit Warten auf entsprechende Antwort des Antrieb (z.B. von Achse 1):

COB-ID	Daten	Kommentar
201	06 00	Shutdown command
181	31 02	State Ready_to_switch_on
201	07 00	Switch_on command
181	33 02	State Switched_on
201	0F 00	Enable_operation command
181	37 02	State Operation_enabled
201	1F 00	Enable IP-mode
181	37 12	IP-mode enable

Diese Konfiguration gibt nun eine zyklische Sequenz frei:



z.B. 2 Achsen

t cycle 1 ms pro Achse bei 1 MBaud

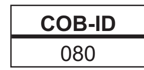
RPDO 2 kann jetzt für die Übertragung von Trajektorien Daten für beide Achsen verwendet werden, z.B.:

COB-ID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
301	F4	01	00	00	E8	03	00	00

Hier erhält die erste Achse den Trajektorien-Wert 500 Inkremente (Bytes 0 bis 3) und die zweite Achse erhält den Wert 1000 Inkremente (Bytes 4 bis 7).

Die Achsen akzeptieren diese Werte und die Positionierung wird durchgeführt, wenn das nächste SYNC Telegramm empfangen wird.

SYNC Telegramm



COB-ID
080

Danach senden beide Achsen ihre inkrementelle Position und den Inhalt ihres Status Registers zurück, wenn das SYNC Objekt mit der COB-ID für das 2. TPDO empfangen wird:

COB-ID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Kommentar
181	23	01	00	00	00	00	03	44	Position + Hersteller Status Register für Achse 1
182	A5	02	00	00	00	00	03	44	Position + Hersteller Status Register für Achse 2

Wenn während des Vorgangs ein Fehler auftritt, sendet die betroffene Achse eine *Emergency* Meldung, die so aussehen könnte:

Emergency Objekt

COB-ID	Emergency error		Error register	Kategorie		
	Low	High				
081	10	43	08	01	00 00 00 00	Motor Temperatur, Temperatur, herstellerspezifisch
081	00	00	08	00	00 00 00 00	

5.3 Stichwortverzeichnis

Vertrieb und Applikation

Wir bieten Ihnen einen kompetenten und schnellen Service. Wählen Sie das zuständige regionale Vertriebsbüro in Deutschland oder kontaktieren Sie den europäischen oder nordamerikanischen Kundendienst.

Deutschland

Danaher Motion GmbH
Vertriebs- & Applikationszentrum Nord
Wacholderstr. 40-42
D-40489 Düsseldorf
Internet www.kollmorgen.com
E-Mail vertrieb.nord@kollmorgen.com
Tel.: +49(0)203 - 99 79 - 250
Fax: +49(0)203 - 99 79 - 3315

Danaher Motion GmbH
Vertriebs- & Applikationszentrum Süd
Brückenfeldstr. 26/1
D-75015 Bretten
Internet www.kollmorgen.com
E-Mail vertrieb.sued@kollmorgen.com
Tel.: +49 (0) 7252 96462 0
Fax: +49 (0) 203 9979 3317

Danaher Motion GmbH
Vertriebsbüro Süd
Münzgasse 6
D-72379 Hechingen
Internet www.kollmorgen.com
E-Mail vertrieb.sued@kollmorgen.com
Tel.: +49 (0) 7471 99705-10
Fax: +49 (0) 203 9979 3316

Europa

Kollmorgen Kundendienst Europa
Internet www.kollmorgen.com
E-Mail technik@kollmorgen.com
Tel.: +49(0)203 - 99 79 - 0
Fax: +49(0)203 - 99 79 - 155

Nordamerika

Kollmorgen Kundendienst Nord Amerika
Internet www.kollmorgen.com
E-Mail support@kollmorgen.com
Tel.: +1 - 540 - 633 - 3545
Fax: +1 - 540 - 639 - 4162

KOLLMORGEN®